



TUGAS AKHIR - MO141326

**PENGEMBANGAN *SOFTWARE* ANALISA KEANDALAN
STRUKTUR DENGAN MODEL DINAMIS BERBASIS
MONTE CARLO FINITE ELEMENT METHOD PADA
STRUKTUR SEDERHANA 2 DIMENSI**

MUHAMMAD ICHSAN

NRP. 4313100016

Dosen Pembimbing

Ir. Handayanu, M.Sc., Ph.D.

Agro Wisudawan., S.T., M.T.

DEPARTEMEN TEKNIK KELAUTAN

FAKULTAS TEKNOLOGI KELAUTAN

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2017



FINAL PROJECT - MO141326

Development of Structural Dynamic Analysis Software of Truss Structure Based on Monte Carlo Finite Element Method for Simple Structure 2 Dimension

MUHAMMAD ICHSAN

NRP. 4313100016

Supervisors

Ir. Handayanu, M.Sc., Ph.D.

Agro Wisudawan., S.T., M.T.

DEPARTMENT OF OCEAN ENGINEERING

FACULTY OF MARINE TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2017

**PENGEMBANGAN SOFTWARE ANALISA KEANDALAN
STRUKTUR DENGAN MODEL DINAMIS BERBASIS MONTE
CARLO FINITE ELEMENT METHOD PADA STRUKTUR
SEDERHANA 2 DIMENSI**

HALAMAN PENGESAHAN TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik pada

Progran Studi S-1 Departemen Teknik Kelautan

Fakultas Teknologi Kelautan

Institut Teknologi Sepuluh Nopember

Oleh :

MUHAMMAD ICHSAN

NRP. 4313 100 016

Disetujui oleh:

1. Ir. Handayanu, M.Sc., Ph.D. (Pembimbing 1)

2. Agro Wisudawan, S.T., M.T. (Pembimbing 2)

3. Dr. Eng., Yeyes Mulyadi, S.T., M.Sc. (Penguji 1)

4. Silvianita, S.T., M.Sc., Ph.D. (Penguji 2)

5. Wimala L. Dhanista, ST., MT. (Penguji 3)

Surabaya, Desember 2017

PENGEMBANGAN SOFTWARE ANALISA KEANDALAN STRUKTUR DENGAN MODEL DINAMIS BERBASIS MONTE CARLO FINITE ELEMENT METHOD PADA STRUKTUR SEDERHANA 2 DIMENSI

Nama Mahasiswa : Muhammad Ichsan
NRP : 4313 100 016
Dosen Pembimbing : Ir. Handayanu, M.Sc., Ph.D
Agro Wisudawan, S.T., M.T

ABSTRAK

Analisa keandalan struktur semakin sering digunakan beriringan dengan kondisi perancangan struktur bangunan lepas pantai yang memiliki ketidakpastian cukup tinggi, selain itu pertimbangan sisi ekonomis bangunan terhadap lingkungan laut. Konfigurasi struktur yang rumit menjadi perhatian sehingga muncul Metode Elemen Hingga. Dua permasalahan antara keandalan dan kerumitan struktur ini menghasilkan metoda analisa *Monte Carlo Finite Elemen Method* (MCFEM). Pada bagian pertama dari penelitian ini dilakukan perbandingan hasil analisa deterministik antara *software* yang dibuat pada *Matlab* dengan *SAP 2000*. Pada kasus ini struktur *Plane Truss* sederhana dengan derajat kebebasan banyak dikenai beban Model Dinamis pada tumpuan struktur. Pada bagian kedua setelah mendapatkan hasil yang menyerupai *software SAP 2000*, dilanjutkan penerapan algoritma MCFEM pada algoritma analisa deterministik yang sebelumnya telah dibuat. Hasil dari penelitian ini, memuat prosedur baru berupa algoritma perhitungan analisa keandalan dengan menggabungkan Simulasi *Monte Carlo* dan Metode Elemen hingga pada Model Dinamis. Selain itu, dengan *software* yang berbasis *Matlab* yang telah memiliki keakuratan seperti *software* komersial dapat digunakan untuk menganalisa model struktur yang lebih kompleks. Setelah dilakukan proses analisa keandalan sebanyak 100000 kali simulasi selama 4147 detik, menunjukkan hasil yang andal dengan *probability of failure* adalah 0 atau keandalan sama dengan 1.

Kata kunci: Analisa Keandalan, Dinamika Struktur, Metode Elemen Hingga, Monte Carlo, dan *Software Matlab*

Development of Structural Dynamic Analysis Software of Truss Structure Based on Monte Carlo Finite Element Method for Simple Structure 2 Dimension

Name : Muhammad Ichsan
NRP : 4313 100 016
Dosen Pembimbing : Ir. Handayanu, M.Sc., Ph.D
Agro Wisudawan, S.T., M.T

ABSTRACT

Structural reliability analysis is increasingly used in conjunction with the design of offshore structures which have high uncertainty, Further, the consideration of the economic side of structures in the offshore environment. The complex configuration of the structure become concern so Finite Element Method appears. Two problems between reliability and complexity of these structures resulted in Monte Carlo Finite Element Method (MCFEM) method of analysis. In the first part of this research, we will show results comparison of deterministic analysis between software made in Matlab and analysis through SAP 2000. In this case model will assume as a simple Plane Truss structure with multi degrees of freedom appllied dynamic load in the support of structure. The second part after obtaining a good result compare wit SAP 2000, furhter we will apply MCFEM algorithm to the deterministic analysis algorithm which has been made. The results of this research, contains a new procedure in the form of reliability simulation algorithm by combining Monte Carlo Simulation and Finite Element Method in Dynamic Model. In addition, Matlab-based software that already has the accuracy of commercial software can be used to analyze more complex structural models. After a reliability analysis process of 100000 simulations for 4147 seconds, it shows reliable results with probability of failure is 0 or reliability equal to 1.

Key words: Reliability Analysis, Dynamic of Structure, Finite Element Method, Monte Carlo, dan *Matlab Software*

KATA PENGANTAR

Bismillahirrohmanirrohim

Assalamu'alaikum wa rahmatullahi wa barakatuh

Alhamdulillah merupakan limpahan rahmat dan hidayah Allah SWT, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Pengembangan Software Analisa Keandalan Struktur dengan Model Dinamis Berbasis Monte Carlo Finite Element Method pada Struktur Sederhana 2 Dimensi”**.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan program Strata 1 (S-1) di Departemen Teknik Kelautan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya. Bagi penulis, Tugas Akhir ini merupakan karya yang sangat berkesan dalam hidup penulis rasakan selama menempuh perkuliahan di Departemen Teknik Kelautan ITS. Tentunya Tugas Akhir ini tidak luput dari kekurangan, oleh karena itu penulis mengharapkan kritik dan saran. Akhir kata, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi para pembaca.

Billahitaufiq wal hidayah

Wassalamu'alaikum wa rahmatullahi wa barakatuh

Surabaya, Desember 2017

Penulis

UCAPAN TERIMA KASIH

Dalam proses pengerjaan dan penyusunan laporan ini, penulis telah mendapatkan berbagai saran, kritik, dukungan baik secara materil maupun spritual dari berbagai pihak, maka pada kesempatan ini penulis ingin mengucapkan terima kasih terutama kepada:

1. Keluarga besar yaitu Bapak, Ibu, dan kakak yang terus memberikan do'a dan motivasi selama penulis menjalani studi di Institut Teknologi Sepuluh Nopember
2. Bapak Ir. Handayanu, M.Sc., Ph.D dan Bapak Agro Wisudawan, S.T., M.T selaku Dosen Pembimbing Tugas Akhir yang telah memberikan banyak bimbingan sehingga Tugas Akhir ini dapat diselesaikan
3. Bapak Dr. Herman Pratikno. S.T., M.T., selaku Dosen Wali yang selalu mendukung anak-anak walinya untuk segera lulus.
4. Mr. Claudio Pedrazzi, Mr. Bowei Li, Mr. Mahdi Abdeddaim, Kak Abdhy Gazali, S.T., dan Yudnina Nikmatul Hanifah, S.T., who had helped me to solve my theoritical and practical problem.
5. Teman angkatan Teknik Kelautan 2013, saudara Alif, saudara Marsuki dan saudara Alfian yang telah meluangkan banyak waktunya untuk memberikan berbagai macam dukungan serta semua teman-teman yang tidak dapat disebutkan satu persatu.

Terima kasih banyak, semoga kebaikan kalian dibalas oleh Allah SWT.

Surabaya, Desember 2017

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN TUGAS AKHIR	i
ABSTRAK	ii
ABSTRACT	iii
KATA PENGANTAR	iv
UCAPAN TERIMA KASIH	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	x
DAFTAR NOTASI.....	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan penelitian.....	2
1.4. Manfaat	2
1.5. Batasan Masalah.....	3
1.6. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1. Tinjauan Pustaka	5
2.2. Dasar Teori	6
2.2.1. Analisa Keandalan.....	6
2.2.1.1. <i>First Order Second Moment Method</i> (FOSM).....	6
2.2.1.2. Simulasi <i>Monte Carlo</i>	8
2.2.2. Analisa Model Dinamis	12
2.2.2.1. Properti Persamaan Model Dinamis Elemen <i>Plane Truss</i>	14

2.2.2.2. Analisa Modal.....	16
2.2.2.3. Sifat Orthogonalitas.....	18
2.2.2.4. Matriks Redaman Klasik untuk Analisa Dinamis	20
2.2.2.5. Metode <i>Time History</i>	21
BAB III METODOLOGI PENELITIAN.....	25
3.1. Metodologi Penelitian	25
3.2. Prosedur Penelitian.....	26
BAB IV ANALISA DAN PEMBAHASAN	29
4.1. <i>Monte Carlo</i> Finite Elemen <i>Method</i> (MCFEM) pada Model Dinamis	29
4.2. Algoritma dan listing fungsi Aplikasi Simulasi Monte Carlo pada Struktur Model Dinamis	34
4.3. Hasil Perbandingan <i>Software</i>	45
4.3.1. <i>Plane Truss</i> bentuk segitiga.....	45
4.3.2. Struktur <i>Truss</i> 2 Tingkat.....	48
4.4. Penerapan Algoritma Simulasi <i>Monte Carlo</i> pada Algoritma Metode Elemen Hingga Model Dinamis beserta listing yang digunakan	50
4.4.1. Struktur <i>Plane Truss</i> Model Segitiga	50
4.4.2. Struktur <i>Truss</i> 2 Tingkat.....	52
BAB V PENUTUP	55
5.1. Kesimpulan	55
5.2. Saran.....	55
DAFTAR PUSTAKA	57
LAMPIRAN 1 PERBANDINGAN DENGAN SOFTWARE SAP 2000.....	59
LAMPIRAN 2 INPUT DAN OUPUT PADA SOFTWARE MCFEM MODEL DINAMIS	87
BIODATA PENULIS	95

DAFTAR GAMBAR

Gambar 2.1. Prosedur simulasi <i>Monte Carlo</i> (Rosyid, 2007)	10
Gambar 2.2. Teknik Transformasi Invers (Ranganathan, 1990)	11
Gambar 2.3. Batang Cantilever dibawah penaruh (a) beban statis dan (b) beban dinamis (Craig, 2006).....	13
Gambar 2.4. Sistem dinamik yang mengalami gerakan pada tumpuannya (Paz, 1990).....	13
Gambar 2.5. Model MDOF: (a) <i>Shear-frame building model</i> dan (b) <i>Spring-mass model</i> (Craig, 2006).....	14
Gambar 2.6. Member dari <i>Plane Truss</i> yang menunjukkan <i>Displacement</i> nodal dan gaya (M. Paz, 1990)	15
Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir	25
Gambar 4.1. Diagram Penentuan Sampel Variabel Acak (Guoliang et. al., 1993)	31
Gambar 4.2 Algoritma Solusi Persamaan Gerak Struktur Model Dinamis	36
Gambar 4.2 Algoritma Solusi Persamaan Gerak Struktur Model Dinamis (Lanjutan)	37
Gambar 4.3 (a) Ilustrasi Model <i>Plane Truss</i> dan (b) Eksitasi pada pondasi <i>Plane Truss</i> ..	45
Gambar 4.4 Percepatan yang terjadi pada Tumpuan Struktur.....	46
Gambar 4.5 Perbandingan Grafik <i>Displacement</i> pada Node 2 Sumbu-x (a) Matlab dan (b) SAP 2000.....	47
Gambar 4.6 Perbandingan Grafik <i>Displacement</i> pada Node 2 Sumbu-z (a) Matlab dan (b) SAP 2000.....	47
Gambar 4.6 Perbandingan Grafik <i>Displacement</i> pada Node 2 Sumbu-z (a) Matlab dan (b) SAP 2000 (Lanjutan)	48
Gambar 4.7 Model Elemen Truss 2D Hasil simulasi	48
Gambar 4.8 Perbandingan Grafik <i>Displacement</i> pada Node 2 Sumbu-z (a) Matlab dan (b) SAP 2000.....	49
Gambar 4.9 Grafik Hubungan Jumlah Simulasi dengan Keandalan	51
Gambar 4.10 Model Elemen Truss 2D Hasil simulasi	52

Gambar 4.11 Grafik Hubungan Jumlah Simulasi dengan Keandalan	53
--	----

DAFTAR TABEL

Tabel 4.1 Limit untuk <i>Displacement</i> Tingkatan Struktur Elastis (Code for Seismic Design of Building, 2010).....	32
Tabel 4.2 Material Properties Struktur Truss	45
Tabel 4.3 Perbandingan Frekuensi Natural Model pada Matlab dan SAP 2000.....	46
Tabel 4.4 Perbandingan <i>Displacement</i> Maksimum dan Minimum pada Matlab dan SAP 2000.....	46
Tabel 4.5 Properti Material Struktur <i>Truss</i> 2 Tingkat.....	49
Tabel 4.6 Perbandingan Hasil Frekuensi Natural Struktur <i>Plane Truss</i> 2 Tingkat	49
Tabel 4.7 Perbandingan Hasil <i>Displacement</i> Maksimum pada Struktur Truss 2 Tingkat ..	50
Tabel 4.8 <i>Mechanical Properties</i> pada Struktur Plane Truss Model Segitiga	50
Tabel 4.9 Hasil <i>Running</i> Struktur <i>Plane Truss</i> Model Segitiga	51
Tabel 4.10 Material Properties Struktur Truss	52
Tabel 4.11 Menampilkan hasil <i>Displacement</i> terbesar pada nodal 3	53

DAFTAR NOTASI

$f_x(\mathbf{X})$	fungsi kepadatan sendi dari variabel x_1, x_2, \dots, x_n
$d\mathbf{X}$	mewakili $dx_1, dx_2, dx_3, \dots, dx_n$
N	banyaknya siklus simulasi
N_f	banyaknya kegagalan (untuk $MK(X) < 0$)
E	<i>Modulus Elastisitas</i>
A	<i>Luas Penampang</i>
m	<i>inersia massa struktur (lb.sec²/in)</i>
c	<i>redaman struktur (ln.sec/in)</i>
k	<i>kekakuan struktur (lb/in)</i>
\ddot{x}	<i>percepatan (in/s²)</i>
\dot{x}	<i>kecepatan (in/s²)</i>
x	<i>perpindahan (in)</i>
$P(t)$	<i>gaya inersia (lbf)</i>
C	<i>cos θ</i>
S	<i>sin θ</i>
q_i	<i>perpindahan modal (in)</i>
ϕ	<i>matriks moda getar yang telah dinormalisasi</i>
U	<i>perpindahan vektor (in)</i>
$\{r\}$	<i>influence coefficient vector</i>
$[M_m] = [\phi]^T [m] [\phi]$	<i>generalized mass matrix</i>
$[C_d] = [\phi]^T [c] [\phi]$	<i>generalized redaman matrix</i>
$[K_d] = [\phi]^T [k] [\phi]$	<i>generalized stiffness matrix</i>
$A = k^{-1}m$	<i>determinan matriks</i>
$\lambda = \frac{1}{\omega^2}$	<i>lamda atau satu dibagi frekuensi struktur kuadrat</i>
I	<i>matriks identitas</i>

Ω	<i>matriks diagonal yang berisi nilai eigen ω_i^2 sebagai diagonalnya</i>
C	<i>matriks diagonal redaman</i>
ζ	<i>rasio redaman</i>
ω_n	<i>frekuensi natural struktur (rad/s)</i>
$\Gamma_i = \frac{[\phi_i]^T[m][r]}{[\phi_i]^T[m][\phi_j]}$	<i>faktro partisipasi modal</i>
$p(\tau)$	<i>pembebanan dinamik (lbf)</i>
\ddot{q}_{i+1}	<i>percepatan modal pada waktu $i+1$</i>
\dot{q}_{i+1}	<i>kecepatan modal pada waktu $i+1$</i>
q_{i+1}	<i>perpindahan modal pada waktu $i+1$</i>
Δt	<i>selisih waktu</i>

DAFTAR LAMPIRAN

- LAMPIRAN 1 PERBANDINGAN DENGAN SOFTWARE SAP 2000
- LAMPIRAN 2 INPUT DAN OUPUT PADA SOFTWARE MCFEM MODEL
DINAMIS

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1. Latar Belakang

Berdasarkan fakta-fakta lapangan, menunjukkan bahwa variabel yang bersangkutan pada analisa dan desain sistem struktur, seperti beban lingkungan, kekuatan material, dll dikenal sebagai sifat dasar yang acak. Agar dapat menjelaskan perilaku tersebut, statistik atau metode probabilistik seharusnya digunakan (Schueller, 1987).

Pada analisa dan desain tradisional dari suatu struktur terbatas memberikan informasi pada variabel kekuatan dan beban struktur disederhanakan sebagai nilai “maksimum” (beban) dan nilai “minimum” (kekuatan) yang sebagian besar merupakan berdasarkan teori dan informasi secara empiris mapun sebagai keputusan seorang *engineer* yang dipilih (Schueller, 1987).

Beberapa peneliti telah berupaya menurunkan persamaan untuk memprediksi parameter-parameter acak ini melalui pendekatan stokastik (Cherng, 1992). Pendekatan inilah yang akan menjadi sebuah acuan pendekatan baru pada sistem struktur.

Hanya saja kompleksitas geometri struktur tidak mungkin dianalisa dengan menggunakan perhitungan analitis, maka salah satu cara untuk memudahkan perhitungan dibantu dengan komputerisasi yang dikenal dengan simulasi numerik (Chapra, 2009).

Metode numerik yang hari ini banyak digunakan adalah salah satunya Metode Elemen Hingga, penyelesaian menggunakan metode ini terbatas pada parameter-parameter yang sudah ditentukan nilainya (deterministik) (Guoliang et. al, 1993). Maka, untuk mensiasati keterbatasan ini, peneliti mencoba menggabungkan antara analisa metode elemen hingga dan statistik untuk mendapatkan hasil yang efisien dan efektif (Koduru, 2010).

Salah satu metode statistik yang dikembangkan adalah dengan menerapkan analisa *Monte Carlo* pada penyelesaian deterministik. Dengan *Monte Carlo* ini, variabel yang tetap diubah menjadi variabel acak penentuan variabelnya dengan mensimulasikan berdasarkan distribusi peluangnya (Rosyid, 2007).

Keterbatasan masing-masing dari dua metode di atas, melahirkan suatu inovasi sebagai obyek penelitian dan publikasi para ilmuwan. Publikasi ini merupakan penerapan simulasi *Monte Carlo* pada metode elemen hingga. Konsep penggabungan 2 metode ini

dipublikasikan dengan nama *Monte Carlo Finite Element Method* (MCFEM) (Wisudawan, 2013).

Ternyata sebelum itu pula telah dilakukan penelitian serupa yang dikenal dengan *Stochastic Finite Element Method* (SFEM) yang dimana variabelnya bersifat stokastik namun dalam perhitungan jumlah menggunakan analisa deterministik (Haldar & Mahadevan, 1991).

Penelitian MCFEM ini terus berlanjut, salah satunya adalah pada analisa struktur *jacket* APN-A Platform dengan mengevaluasi hasil yang ditemukan dengan menggunakan *software* SACS (Baihaqi, 2014).

Atas latar belakang tersebut penelitian ini akan membahas analisa Dinamis pada struktur *Plane Truss* dengan menggunakan simulasi *Monte Carlo Finite Element Method*.

Dalam analisa komputasi numerik dan desain dalam aplikasi sains dikenal beberapa *software* seperti *Maple*, *Mathematica*, *Maxima*, dan sebagainya. Namun seiring berkembangnya waktu muncullah *Matlab* yang mendominasi pasar. (Campbell, et al., 2010). Adapun pada penelitian kali ini digunakan *software Matlab* versi 2011.

1.2. Perumusan Masalah

Permasalahan yang akan dikaji dalam tugas akhir ini adalah:

- a. Bagaimana hasil analisa model dinamis pada *software Matlab* versi 2011 dibandingkan dengan *software* komersial?
- b. Bagaimana penerapan *Monte Carlo* pada Metode Elemen Hingga model dinamis menggunakan *software Matlab*?

1.3. Tujuan penelitian

Dari perumusan masalah di atas, dapat diambil suatu tujuan dari penelitian yang dilakukan, yaitu:

- a. Mendapatkan *listing* fungsi analisa model dinamis dengan menggunakan *software Matlab* versi 2011.
- b. Mendapatkan *listing* analisa keandalan struktur dengan menerapkan *Monte Carlo* pada perhitungan yang telah dicek keakuratannya dengan *software* komersial.

1.4. Manfaat

Berikut adalah manfaat yang dapat diambil dari penelitian yang dilaksanakan:

- a. Memberikan keuntungan dari segi ekonomis karena kesesuaiannya dengan *software* yang komersial.
- b. Dapat digunakan untuk menghitung keandalan struktur *Plane Truss* dalam kehidupan sehari-hari.

1.5. Batasan Masalah

Untuk mempersempit masalah dan memudahkan perhitungan, maka masalah akan dibatasi dengan beberapa hal sebagai berikut:

- a. Dalam 1 elemen, material properties dan luasan dianggap homogen sepanjang batang.
- b. Input variabel acak berupa variabel Modulus Elastisitas (E), Luas Penampang (A), dan *Displacement* batas ($u(t)_{\text{control}}$).
- c. Rasio redaman yang digunakan sebesar 5%
- d. Moda kegagalan yang digunakan adalah kegagalan akibat *over Displacement*.
- e. Kondisi *pile* diasumsikan dengan tumpuan bebas pada permukaan tanah.

1.6. Sistematika Penulisan

Pada Bab I (satu), penulis menjelaskan latar belakang penelitian yang dilakukan, perumusan masalah, tujuan Bab ini menjelaskan tentang latar belakang penelitian yang akan dilakukan, perumusan masalah, tujuan yang hendak dicapai dalam penulisan tugas akhir, manfaat yang diperoleh, serta ruang lingkup penelitian untuk membatasi analisis yang dilakukan dalam tugas akhir.

Bab II (dua) ini berisi referensi dan juga teori-teori pendukung yang digunakan sebagai acuan atau pedoman dalam menyelesaikan tugas akhir. Referensi tersebut bersumber pada jurnal lokal maupun internasional, literatur, *rules/code* dan juga buku yang berkaitan dengan topik yang dibahas.

Bab III (tiga) ini menjelaskan langkah-langkah pengerjaan tugas akhir dan metodologi yang digunakan.

Bab IV (empat) ini menjelaskan mengenai pembahasan hasil penelitian yang menyangkut masalah algoritma pembuatan *software*, penerapan keandalan struktur dengan membuat listing pada *Matlab*, prosedur analisa hasil perbandingan antara kedua *software*.

Bab V (lima) ini menjelaskan kesimpulan yang berisikan tahap akhir dari penulisan tugas akhir yang membandingkan hasil akhir *software* yang dibuat dengan *software* SAP 2000.

Sehingga, dapat disimpulkan kelayakan listing untuk diterapkan metode monte carlo di dalamnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Sebagian metode telah dipaparkan pada buku *Stochastic Dynamics of Structure* oleh Jie Liang dan Jianbing Chen bahwa selama ini metode elemen hingga telah banyak dipublikasikan dalam terminologi deterministik, padahal kenyataannya metode elemen hingga ini dapat diuraikan pada isu peluang kejadian variabel acak. Variabel acak yang tidak terkontrol selalu ada pada pembuatan konstruksi sistem yang nyata (Clough, 2003).

Untuk mengakomodasi ketidakpastian pada variabel acak maka diperlukan analisa proses stokastik dimana melalui proses ini input variabel acak akan menghasilkan variabel *output* yang acak juga. (Jie Li, 2009)

Ada 3 jenis metode dalam menentukan ketidakpastian respon struktur yaitu Metode Monte Carlo, Teknologi Pertubasi, dan Ekspansi Polinomial Orthogonal yang dimana ke tiga metode ini diterapkan pada analisa metode elemen hingga. (Jie Li, 2009)

Pada penelitian lebih lanjut untuk mempermudah perhitungan pada konfigurasi struktur yang kompleks, metode yang digunakan dengan menggabungkan antara analisa keandalan dan elemen hingga.

Simulasi *Monte Carlo* adalah jenis simulasi yang mengandalkan pengambilan sampel berulang secara acak dan mengandalkan analisa statistik untuk menghitung sesuatu. (Raychaudhuri, 2008)

Metode elemen hingga sendiri berfungsi untuk melakukan analisa numerik yang masalah teknik dan problem matematis dari suatu kenampakan fisiknya. Tipe masalah yang ditangani ini terbagi dua kelompok, yaitu kelompok analisa struktur dan kelompok masalah-masalah non struktur.

Sehingga peneliti menghasilkan metode *Monte Carlo Finite Elemen Method*, dimana metode ini dilakukan guna untuk menentukan keandalan struktur dengan melakukan identifikasi semua distribusi yang berkaitan dengan keandalan, membuat data acak, menentukan sampel, menghitung dengan metode elemen hingga dan terakhir dengan fungsi keandalan (Guoliang et. Al, 1993).

Sebuah metode dikembangkan dengan menggunakan simulasi *Monte Carlo Finite Elemen Method* pada keandalan struktur *Plane Truss* (Wisudawan et. al, 2013). Penelitian ini

pengembangan pada tingkatan struktur yang lebih kompleks yang dilakukan pada struktur Jacket.

Selain itu pada penelitian lain oleh *Monte Carlo Finite Elemen Method* berbasis *Matlab* juga dilakukan pada analisa keandalan struktur *space frame* pada *APN-A Platform* menunjukkan hasil yang cukup baik setelah divalidasi dengan *software SACS* (Baihaqi, 2015).

2.2. Dasar Teori

2.2.1. Analisa Keandalan

2.2.1.1. *First Order Second Moment Method* (FOSM)

Pada analisa FOSM, orde pertama pendekatan *taylor series* dari fungsi *limit state* adalah fungsi yang digunakan dan hanya momen kedua statistik dari variabel acak yang digunakan untuk mendapatkan *probability of failure* (P_f) (Datta, 2010). Dalam bentuk original, Cornell menggunakan 2 random variabel untuk menurunkan persamaan. Fungsi limit state dinyatakan sebagai berikut:

$$Z = R - S \quad (2.1)$$

Dengan mengasumsikan R dan S adalah variabel acak dengan distribusi normal statis independen, Z juga dinyatakan sebagai distribusi normal. Mean dan kovarian diperoleh dengan:

$$\mu_Z = \mu_R - \mu_S \text{ dan } \sigma_Z^2 = \sigma_R^2 - \sigma_S^2 \quad (2.2)$$

Probabilitasn kegagalan dinyatakan dnegan:

$$P_f = P[Z < 0] = P[(R - S) < 0] \quad (2.3)$$

Jika Z adalah variasi normal, maka akan mudah menyatakan P_f sebagaimana persamana dibawah ini:

$$P_f = \phi\left(-\frac{\mu_Z}{\sigma_Z}\right) \quad (2.4)$$

ϕ adalah fungsi distribusi kumulatif untuk variabel standar normal.

Rasio dari $\frac{\mu_z}{\sigma_z}$ dapat dituliskan dengan variabel β dan familiar dengan sebutan indeks keandalan (indeks keamanan) pada teori keandalan. Kemudian P_f lebih populer dengan persamaan berikut:

$$P_f = \phi(-\beta) \quad (2.5)$$

Jika variabel R dan S adalah distribusi *log-normal*, maka fungsi *limit state* dinyatakan sebagai berikut:

$$Z = \ln\left(\frac{R}{S}\right) \quad (2.6)$$

Z sekali lagi adalah variabel normal, dan probabilitas kegagalan dapat dinyatakan sebagai berikut:

$$P_f = 1 - \phi\left\{ \frac{\frac{\mu_z}{\mu_s} \sqrt{\frac{1 + \delta_s^2}{1 + \delta_R^2}}}{\sqrt{\ln(1 + \delta_R^2) \ln(1 + \delta_S^2)}} \right\} \quad (2.7)$$

δ_R dan δ_S adalah koefisien variasi dari R dan S.

Formulasi diatas dapat berlaku secara umum untuk banyak variabel acak, dituliskan sebagai vektor \mathbf{X} . Dimana fungsi kinerja dapat dituliskan sebagai berikut:

$$Z = MK(\mathbf{X}) \quad (2.8)$$

Ekspansi *Taylor Series* dari fungsi kinerja mengenai mean dinyatakan sebagai berikut:

$$Z = MK(\mathbf{X}) + \sum_{i=1}^n \frac{\partial MK}{\partial x_i} (x_i - \bar{x}_i) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 MK}{\partial x_i \partial x_j} (x_i - \bar{x}_i)(x_j - \bar{x}_j) + \dots \quad (2.9)$$

Dimana \bar{x}_i adalah mean dari variabel x_i . Singkatan dari persamaan series pada istilah linier, pendekatan *first-order* dari mean dan varian dari Z ditentukan sebagai berikut:

$$\bar{Z} = MK(\bar{\mathbf{X}}) \quad (2.10)$$

$$\sigma_z^2 = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 MK}{\partial x_i \partial x_j} (x_i, x_j) \quad (2.11)$$

Dimana $\text{cov}(x_i, x_j)$ adalah kovarian dari x_i dan x_j . jika variabel diasumsikan sebagai statis independen, maka

$$\sigma_z^2 = \sum_{i=1}^n \left(\frac{\partial^2 MK}{\partial x_i} \right)^2 \sigma_{x_i}^2 \quad (2.12)$$

Turunan parsial dibutuhkan untuk mendapatkan hasil dari mean di atas.

2.2.1.2. Simulasi *Monte Carlo*

Karakteristik pokok simulasi *Monte Carlo* adalah bahwa simulasi ini, suatu set nilai tiap-tiap variabel (satu nilai untuk setiap variabel) dari suatu sistem disimulasikan berdasarkan distribusi peluangnya, misalnya berdasarkan fungsi kerapatan peluang tiap-tiap variabel tersebut.

Prosedur ini diulang-ulang sampai diperoleh sekumpulan nilai dari respon atau kinerja sistem tersebut, sehingga pada akhir simulasi akan diperoleh sekumpulan data respon atau kinerja sistem. Sekumpulan data ini dapat dianggap sebagai sampel data. Dengan analisis statistik dapat dilakukan untuk menentukan rata-rata, simpangan baku, bahkan distribusi dari respon atau kinerja sistem tersebut.

Unsur pokok yang diperlukan didalam simulasi *Monte Carlo* adalah sebuah *random number generator*. Hal ini dikarenakan secara teknis, prinsip dasar metode simultan *Monte Carlo* sebenarnya adalah sampling numerik dengan bantuan *random number generator*, dimana simulasi dilakukan dengan mengambil beberapa sampel dari peubah acak berdasarkan distribusi peluang peubahan acak tersebut.

Dengan bantuan komputer yang semakin besar kemampuan mengolah data, simulasi *Monte Carlo* dapat merupakan simulasi alternatif yang lebih bermanfaat, bahkan untuk sistem yang kompleks. Simulasi dapat dilakukan tanpa harus melakukan penurunan parsial dan menyelesaikan suatu sistem persamaan simultan sebagaimana *FOSM* dan *AFOSM*. (Rosyid, 2007).

Hal ini berarti simulasi *Monte Carlo* mensyaratkan bahwa distribusi peluang dari peubahan acak yang terlibat didalam sistem yang sedang dipelajari telah diketahui atau dapat diasumsikan. Sampel yang telah diambil tersebut dipakai sebagai masukan ke dalam persamaan fungsi kinerja $MK(X)$ dan harga $MK(X)$ kemudian dihitung.

Fungsi kerja atau fungsi *limit state* diberikan oleh $MK(X)=0$. Sedangkan *probability of failure* P_f diperoleh sebagai berikut:

$$P_f = \iint \dots \int_{MK(X) \leq 0} f_x(X) dX \quad (2.13)$$

Variabel dasar dari simulasi *monte carlo* berdasarkan karakteristik probabilitas dan memberikan variabel tadi dalam fungsi limit state. *Probability of failure* dapat dikomputasikan dengan pendekatan sebagai berikut:

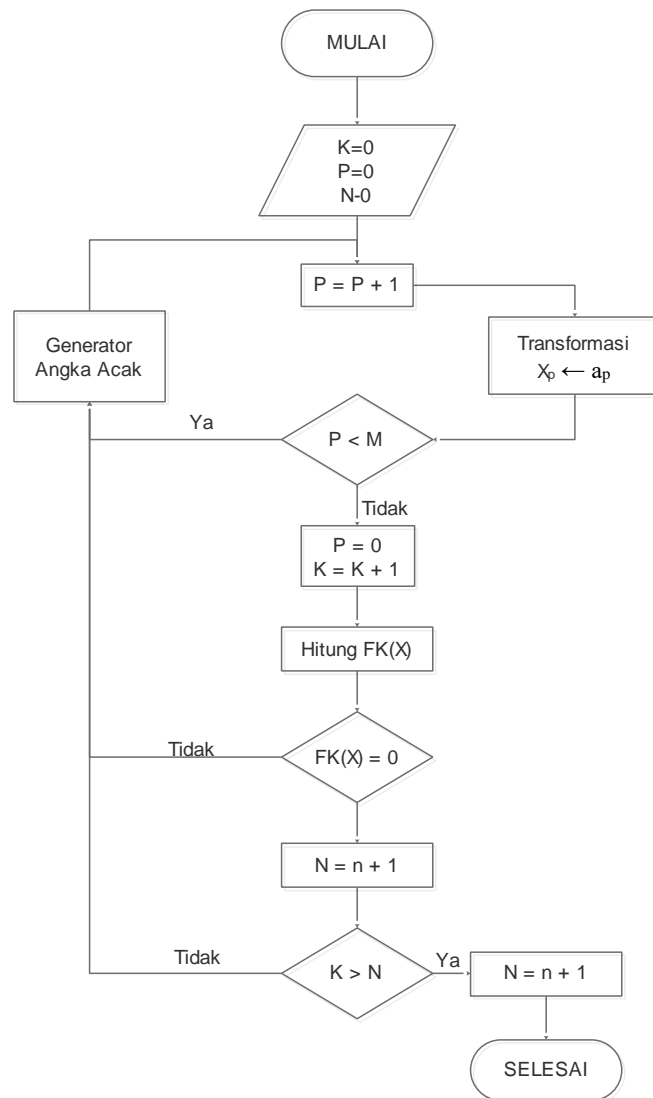
$$P_f = \frac{N_f}{N} \quad (2.14)$$

Untuk suatu fungsi kinerja tertentu, misalnya setiap kali $MK(X) < 0$ maka sistem/komponen yang ditinjau dianggap gagal, jika jumlah sampel tersebut adalah N atau replika sejumlah (N) maka dapat dicatat kejadian $MK(X) < 0$ sejumlah N_f kali. Dengan demikian, peluang kegagalan sistem/komponen yang sedang ditinjau adalah rasio antara jumlah kejadian gagal dengan banyaknya sampel atau replikasi sebagaimana fungsi di atas.

Estimasi *probability of failure* tergantung dari banyaknya simulasi yang dilakukan. Untuk hasil yang cukup akurat, maka diperlukan semakin banyak pula simulasi yang dilakukan. Oleh karena itu, digunakan pendekatan komputasi varians dari *probability of failure*.

Sebagaimana pada gambar 2.1. dapat dilihat bahwa langkah-langkah yang ditempuh adalah sebagai berikut:

- a. Menentukan fungsi kinerja struktur yang akan disimulasikan.
- b. Memanggil angka acak dari *Random Number Generator*.
- c. Mentransformasi RNG yang merupakan bilangan acak, menjadi nilai peubah bilangan acak yang merupakan salah satu komponen dari fungsi kinerja yang telah ditetapkan.
- d. Menghitung fungsi kerja, jika $MK(X) < 0$ maka struktur gagal.
- e. Ulangi prosedur 1-4 sebanyak yang diinginkan.
- f. Catat berapa kali struktur mengalami kegagalan.



Gambar 2.1. Prosedur simulasi *Monte Carlo* (Rosyid, 2007)

Dimana,

- K : penghitung jumlah eksperimen atau sampling
- M : jumlah peubah dasar dari bilangan random menjadi variabel yang diinginkan
- n : penghitung kondisi gagal ketika fungsi keja $MK(X) < 0$ sehingga struktur dinyatakan gagal.

- Teknik Transformasi Invers

Berdasarkan fungsi distribusi kumulatif atau *cumulative distribution function*, $F_Y(y)$, merupakan distribusi yang akan disimulasikan. $F_Y(y)$ didefinisikan dengan interval (0, 1). Berdasarkan variasi distribusi seragam atau *uniform distribution* V , dimana didefinisikan juga dengan interval (0, 1).

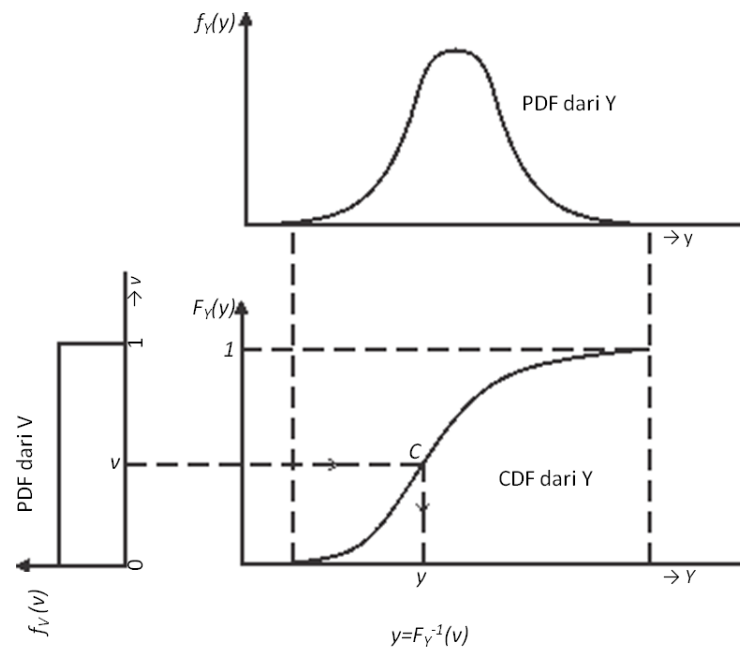
Keluarkan nilai v untuk simpangan acak standar seragam atau *standard uniform random deviate*. Untuk variasi standar seragam V , probabilitas kumulatif dari $V \leq v$ adalah sama dengan v atau dapat dituliskan sebagai berikut:

$$F_Y(y) = v \quad (2.15)$$

Kemudian y secara unik dijealskan dengan relasi

$$F_Y(y) = v \leftrightarrow y = F_Y^{-1}(v) \quad (2.16)$$

Secara grafik dapat ditunjukkan pada gambar . Fungsi kerapatan peluang (FKP) dari V pada sebelah kiri. Menghasilkan angka acak seragam v diproyeksi pada kurva CDF dari suatu fungsi Y . Pada titik C pada kurva diproyeksikan kebawah di axis horizontal untuk mendapatkan nilai sama. Oleh karena itu, $y = F_Y^{-1}(v)$ adalah variasi yang didapatkan dari hasil distribusi Y .



Gambar 2.2. Teknik Transformasi Invers (Ranganathan, 1990)

Ketika invers $F_Y(y)$ contohnya $F_Y(y)$, tidak ada atau sangat sulit dipraktikkan, maka dapat digunakan teknik terima-tolak atau *rejection technique*, metode komposisi, metode aproksimasi. Oleh sebab itu, prosedur yang disarankan untuk penggambaran k th set dari suatu nilai input y_{ik} dari distribusi yang sama dengan F_{Yi} didapatkan pertama sebuah set dari n angka acak, v_{ik} , dengan fungsi kepadatan seragam dengan rentang $0 \leq v \leq 1$. Nilai dari y_{ik} dapat dituliskan sebagai berikut:

$$y_{ik} = F_{Yi}^{-1}(v_{ik}) \quad (2.17)$$

- Transformasi Distribusi Normal

Untuk kasus distribusi normal yang digunakan adalah teknik *Box dan Muller* untuk menghasilkan distribusi normal (Ranganathan, 1990). Dimana, simpangan standar normal didapatkan dengan menghasilkan dua angka acak uniform v_1 dan v_2 (dengan rentang fungsi kerapatan peluang *uniform* adalah 0 dan 1) pada saat itu. Kemudian untuk mendapatkan variasi standar normal diberikan dengan persamaan di bawah ini:

$$u_1 = [2 \cdot \ln 1 / v_1]^{1/2} \cos(2\pi \cdot v_2) \quad (2.18a)$$

$$u_2 = [2 \cdot \ln 1 / v_1]^{1/2} \sin(2\pi \cdot v_2) \quad (2.18b)$$

Kemudian untuk mengubah variasi standar normal di atas menjadi distribusi normal dari suatu persamaan Y diikuti dengan distribusi normal dengan mean μ dan standar deviasi σ . Pertama dengan dengan mengeluarkan hasil nilai v_1 dan v_2 pada rentang 0 dan 1. Kemudian, dari persamaan 2.18 variasi standar normal dihubungkan dengan persamaan di bawah ini:

$$\frac{Y - \mu}{\sigma} = U \quad (2.19)$$

Dimana U adalah variasi standar normal. Sejak kita menemukan dua variasi normal y_1 dan y_2 menggunakan persamaan 2.18 dan 2.19 atau dituliskan dengan:

$$y_1 = \sigma \cdot u_1 + \mu \quad (2.20a)$$

$$y_2 = \sigma \cdot u_2 + \mu \quad (2.20b)$$

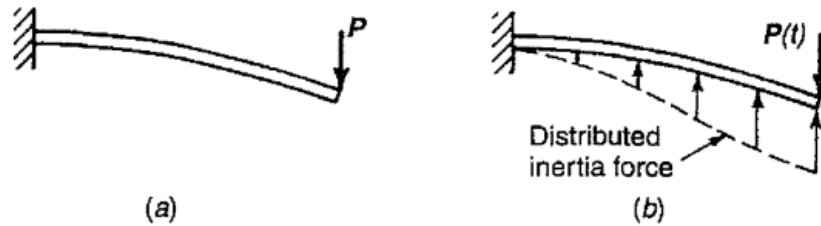
Sehingga persamaan dapat dituliskan dengan:

$$y_1 = \mu + \sigma \cdot [2 \cdot \ln 1 / v_1]^{1/2} \cos(2\pi \cdot v_2) \quad (2.21a)$$

$$y_2 = \mu + \sigma \cdot [2 \cdot \ln 1 / v_1]^{1/2} \sin(2\pi \cdot v_2) \quad (2.22b)$$

2.2.2. Analisa Model Dinamis

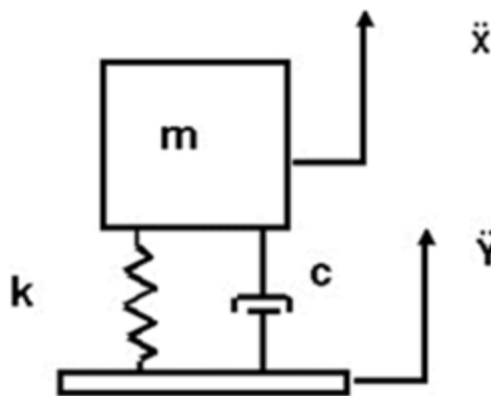
Beban dinamis adalah salah satu besaran, arah, atau aplikasi titik yang bervariasi terhadap waktu. Hasil variasi waktu terhadap *Displacement* dan tegangan merupakan respon dari beban dinamis (Craig, 2006).



Gambar 2.3. Batang Cantilever dibawah penaruh (a) beban statis dan (b) beban dinamis (Craig, 2006)

Gambar 2.3 adalah eksitasi yang diterima oleh pengaruh beban dinamis bervariasi terhadap waktu dan beban yang ada menghasilkan distribusi gaya inersia disepanjang batang. Berbeda dengan beban statis yang hanya menyebabkan defleksi dan tegangan internal.

Respon dinamik dari suatu struktur tidak hanya diakibatkan oleh beban dinamik yang bekerja pada struktur, tetapi dapat juga disebabkan oleh pergerakan tumpuan struktur, misalnya struktur yang menerima beban gempa atau ledakan pada pondasinya (Paz, 1990).



Gambar 2.4. Sistem dinamik yang mengalami gerakan pada tumpuannya (Paz, 1990)

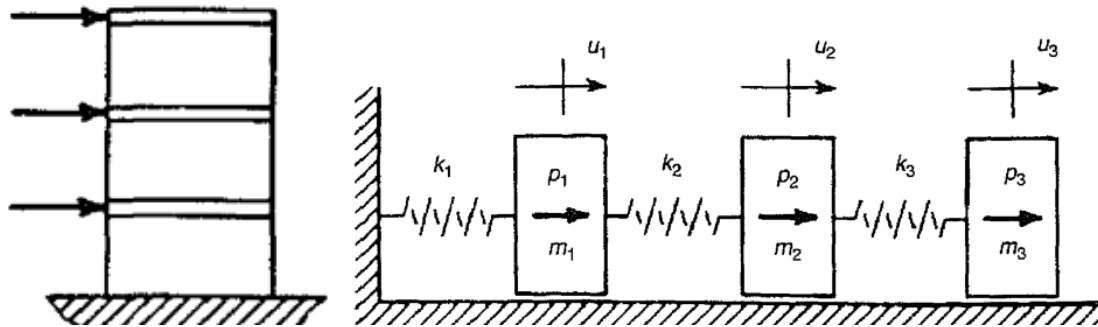
Dalam perencanaan struktur yang sebenarnya, akan dicari sistem dengan konfigurasi massa dan kekakuan yang dapat memberikan perpindahan massa sekecil mungkin. Hal ini dapat dicapai dengan mengatur harga-harga massa, kekakuan dan redamannya.

Gaya yang bekerja pada sistem dalam hal ini bukanlah gaya yang bekerja pada massa secara langsung, tetapi merupakan gaya inersia yang diakibatkan oleh gerakan tumpuannya.

Perhatikan gambar 2.3, tumpuan struktur mengalami gerakan harmonik dengan frekuensi eksitasi Ω dan amplitudo gerakan X_s .

$$x_s(t) = X_s \sin \Omega t \quad (2.23)$$

Namun, fakta observasi menyatakan bahwa sistem struktur tidak selalu memiliki derajat kebebasan yang tunggal ataupun 2 saja melainkan juga memiliki derajat kebebasan banyak atau dikenal dengan istilah *Multi Degree of Freedom* (MDOF).



Gambar 2.5. Model MDOF: (a) *Shear-frame building model* dan (b) *Spring-mass model* (Craig, 2006)

Secara umum persamaan gerak pada sistem struktur dengan derajat banyak dapat dituliskan sebagai berikut:

$$[m]\{\ddot{x}(t)\} + [c]\{\dot{x}(t)\} + [k]\{x(t)\} = P(t) \quad (2.24)$$

Untuk memecahkan persamaan di atas yang saling bergantung (*coupled*) menjadi tidak saling bergantung (*uncoupled*) digunakan metode analisa modal yang akan dibahas selanjutnya.

2.2.2.1. Properti Persamaan Model Dinamis Elemen *Plane Truss*

Dalam menganalisa suatu sistem struktur rangka batang dua dimensi atau *Plane Truss* dapat digunakan metode matriks yang hari ini dikembangkan dengan menggunakan bantuan komputer (Setiawan, 2015).

Pada analisa statis *Plane Truss*, member dianggap sebagai batang yang terhubung dengan pin pada ujung-ujungnya, hal ini mempermudah dalam menetapkan gaya yang bekerja pada setiap titik simpul (nodal).

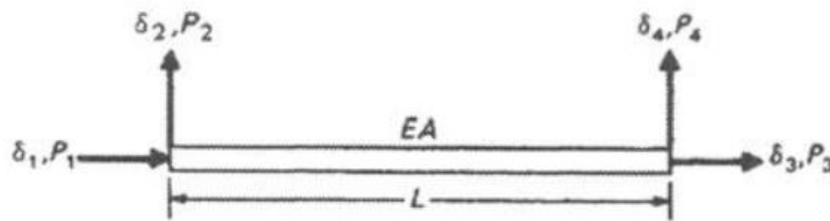
Ketika *Plane Truss* dianggap sebagai struktur yang bekerja secara dinamis, situasi sederhana berupa perkiraan tegangan aksial pada member tidak lagi menjadi satu-satunya yang diperhitungkan. Gaya inersia pada batang juga akan bekerja disepanjang batang, secara umum akan menghasilkan moment lentur disamping tegangan aksial dan dalam kondisi ini bending momen pada ujung batang *Plane Truss* tetap bernilai nol karena tidak adanya momen eksternal pada nodal.

Metode kekakuan dinamis pada analisa truss sebenarnya dikembangkan dari kasus struktur portal untuk menentukan relasi dasar antara gaya eksternal, gaya elastis, gaya redaman, gaya inersia dan menghasilkan perpindahan, *velocity*, dan *acceleration* pada koordinat nodal, yang ditentukan oleh konfigurasi struktur seperti kekakuan, *redaman*, dan matriks massa pada *truss* (Paz, 1990).

Pada sistem *Plane Truss* semua beban bekerja pada nodal, sehingga tidak timbul momen internal. Orientasi dari masing-masing elemen batang harus ditransformasikan ke sumbu global koordinat, sehingga diperlukan adanya matriks transformasi.

Untuk mentransformasikan persamaan dengan koordinat lokal menjadi global dibutuhkan matriks transformasi (T), yaitu sebagai berikut:

$$[T] = \begin{bmatrix} C & S & 0 & 0 \\ -S & C & 0 & 0 \\ 0 & 0 & C & S \\ 0 & 0 & -S & C \end{bmatrix} \quad (2.25)$$



Gambar 2.6. Member dari *Plane Truss* yang menunjukkan *Displacement* nodal dan gaya (M. Paz, 1990)

Sebuah member *Plane Truss* diketahui terdiri dari 2 DOF yang ada pada setiap nodalnya akan mempunyai koordinat nodal secara keseluruhan (gambar 2.6). Untuk lendutan kecil, dapat dianggap bahwa hubungan gaya perpindahan untuk koordinat nodal sepanjang sumbu dari anggota struktur (koordinat 1 dan 3 pada gambar 2.6) tidak dipengaruhi perpindahan melintang koordinat nodal 2 atau 4. Anggapan ini ekuivalen dengan pernyataan bahwa perpindahan menurut koordinat nodal 1 atau 3 tak akan menimbulkan gaya menurut arah koordinat nodal 2 atau 4 ataupun sebaliknya.

Koefisien kekakuan dan koefisien massa sehubungan dnegan koordinat nodal aksial. Adapun yang digunakan sebagaimana gambar 2.6 adalah

$$k_{11} = k_{33} = \frac{AE}{L} \quad k_{13} = k_{31} = -\frac{AE}{L} \quad (2.26a)$$

$$m_{11} = m_{33} = \frac{\bar{m}L}{2} \quad m_{13} = m_{31} = \frac{\bar{m}L}{2} \quad (2.26b)$$

Dimana \bar{m} adalah massa per satuan panjang.

Sehingga, matriks kekakuan lokal *Plane Truss* akan berukuran $2_n \times 2_n$, tergantung banyaknya member yang dirangkaiakan pada *Plane Truss* tersebut, hal ini dapat kita lihat pada gambar 2.6. Selain itu telah diketahui bahwa elemen *Plane Truss* memiliki modulus elastisitas E, luas penampang A, dan panjang L. Matriks kekakuan lokal k dapat dituliskan dengan menyusun koefisien kekuan satu struktur sebagai:

$$[\tilde{k}] = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.27a)$$

Untuk mentransformasi matriks kekakuan lokal menjadi lokal adalah dengan mengalikan dengan matriks transformasi,

$$k = T^T \tilde{k} T \quad (2.27b)$$

Pada matriks massa (M) terdapat 2 jenis penggabungan pada pemodelan struktur portal yaitu *Lumped Mass Matrix* dan *Consistent Mass Matrix*. Prosedur paling sederhana untuk menetapkan sifat massa setiap struktur adalah dengan mengasumsikna sebagai *Lumped Mass Matrix*, didapatkan dengan menggunakan fungsi perpindahan statis dengan prinsip kerja virtual. Dengan mengkordinat lokal adalah sebagai berikut:

$$[\tilde{m}] = \frac{\rho AL}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.28a)$$

Sehingga untuk mentransformasi matriks massa lokal menjadi lokal adalah dengan mengalikan pada matriks transformasi,

$$m = T^T \tilde{m} T \quad (2.28b)$$

2.2.2.2. Analisa Modal

Analisa modal atau dikenal juga dengan superposisi modus adalah suatu proses yang ditujukan untuk menentukan karakteristik-karakteristik dinamik. Dimana karakteristik-karakteristik tersebut adalah frekuensi natural, moda getar dan rasio redam.

Ketiga karakteristik dinamik ini didefinisikan oleh properti fisik serta distribusi spasial daripada komponen penyusun sistem struktur.

Selain menentukan karakteristik tadi, analisa modal juga mencakup penggunaan karakteristik-karakteristik tersebut untuk menentukan kekuatan dinamis dari beban $p(t)$.

Setiap persamaan modal adalah bentuk yang sama dengan persamaan gerak untuk sistem SDOF. Dengan demikian, metode solusi dan hasil yang tersedia untuk sistem SDOF dapat disesuaikan untuk mendapatkan solusi $y(t)$ untuk persamaan modal

$$U = \sum_{i=1}^N [\phi_i][q_i] = \sum_{i=1}^N U_i, U_i = [\phi_i][q_i] \quad (2.29)$$

Selanjutnya menggabungkan persamaan lepas (uncoupled) yang sebelumnya persamaan respon sendiri dikalikan dengan ϕ_i^T , dan selain itu kondisi orthogonal diterapkan sehingga membentuk persamaan sebagaimana di bawah ini:

$$[\phi]^T[m][\phi]\{\ddot{q}\} + [\phi]^T[c][\phi]\{\dot{q}\} + [\phi]^T[k][\phi]\{q\} = -[\phi]^T[m]\{\ddot{x}_g(t)\}\{r\} \quad (2.30a)$$

atau disederhanakan dengan menuliskan

$$[M_m]\{\ddot{y}(t)\} + [C_d]\{\dot{y}(t)\} + [K_d]\{y(t)\} = -[\phi]^T[m]\{r\}\{\ddot{x}_g(t)\} \quad (2.30b)$$

atau dapat dinyatakan juga

$$\ddot{y}(t) + 2\omega\xi\dot{y}(t) + \omega^2y(t) = -\gamma\ddot{x}_g(t) \quad (2.30c)$$

Adapun langkah melakukan analisa modal adalah sebagai berikut:

Pertama, yang dilakukan untuk menentukan moda getar atau *mode shape* ϕ , adalah membuat persamaan (2.24) menjadi undamped free vibration,

$$m.\ddot{x} + k.x = 0 \quad (2.31)$$

Dimana, fungsi gerakan harmonik:

$$x = X\cos(\omega t) + X\sin(\omega t)$$

$$\dot{x} = -\omega X\cos(\omega t) + \omega X\sin(\omega t) \quad (2.32a)$$

$$\ddot{x} = -\omega^2 X\cos(\omega t) - \omega^2 X\sin(\omega t)$$

atau

$$x = X$$

$$\dot{x} = -\omega X \quad (2.32b)$$

$$\ddot{x} = -\omega^2 X$$

Dimana, ϕ didapatkan sebanyak kolom yang ditinjau. Sehingga persamaannya dapat dituliskan sebagai berikut:

$$-\omega^2 mX + kX = 0 \quad (2.33a)$$

$$(k - \omega^2 m)X = 0 \quad (2.33b)$$

dengan syarat:

$$X \neq 0 \text{ dan } (k - \omega^2 m) = 0$$

Untuk menyelesaikan persamaan di atas dilakukan dengan mencari nilai eigen dan vector eigen. Ada 2 cara untuk mencari persamaannya yaitu dengan melakukan invers dan determinan, adapun pada penelitian ini digunakan metode invers yang dijelaskan sebagai berikut:

Persamaan (2.31) dikalikan dengan invers dari matrik kekakuan k^{-1} , maka persamaannya menjadi:

$$k^{-1}mX = \frac{1}{\omega^2}IX \quad (2.34a)$$

atau dapat dituliskan

$$[A - \lambda I]X = 0 \quad (2.34b)$$

Persamaan (2.34a) dikenal dengan persamaan karakteristik untuk sistem tersebut. Persamaan ini dapat diperluas menjadi polynomial dalam λ dan n derajat kebebasan:

$$\lambda'' + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \dots + a_n = 0$$

Akar-akar dari persamaan karakteristik ini disebut dengan nilai eigen atau frekuensi natural dari sistem tanpa redaman dapat diperoleh dari hubungan:

$$\omega_i = \sqrt{1/\lambda_i} \quad (2.34c)$$

Dengan mensubstitusikan λ_i ke dalam persamaan (2.34a) maka akan didapatkan mode getar $\{X\}$ yang biasa juga disebut sebagai vektor eigen. vektor-vektor eigen menggambarkan pola-pola deformasi dari struktur pada setiap frekuensi naturalnya.

2.2.2.3. Sifat Orthogonalitas

Harga dari moda getar X_i adalah harga relatif, elemen-elemen dari moda getar menunjukkan perbandingan satu dengan lainnya. Untuk mendapatkan besarnya harga moda getar yang umum, moda getar yang didapat dari hasil perhitungan problem eigen harus dinormalisasi dengan sifat orthogonalis mode.

Terdapat beberapa cara dalam menetapkan sifat orthogonalis mode seperti:

- Menggunakan Teorema Betti,
- Untuk sistem berderajat kebebasan- n yang memiliki massa diagonal, sifat orthogonal antara 2 mode i dan j dapat di ekspresikan dengan:

$$M = \sum_{k=1}^1 m_k (X_{ki})(X_{kj}) = 0 \rightarrow i \neq j \quad (2.35)$$

- Untuk sistem berderajat kebebasan-n yang memiliki massa umum, sifat orthogonal antara 2 mode i dan j dapat di ekspresikan dengan:

$$\{X\}_i^T [M] \{X\}_j = 0 \rightarrow i \neq j \quad (2.36)$$

- Elemen teratas atau terbawah dari setiap mode getar dibuat sama dengan satu satuan, elemen-elemen lainnya adalah merupakan perbandingan terhadap elemen teratas atau terbawah tersebut.
- Elemen tertentu dari setiap mode getar dibuat sama dengan satu-satuan, elemen-elemen lainnya adalah merupakan perbandingan terhadap elemen tertentu tersebut.
- Panjang vector dari mode getar harus sama dengan satu-satuan.

$$X_1^2 + X_2^2 = 1 \quad (2.37)$$

- Untuk penelitian ini digunakan metode dengan mendapatkan nilai mode getar baru persamaan M yang didapatkan menjadi pembagi untuk setiap mode shape yang telah ditemukan di awal melalui proses analisa modal.

$$\phi_{ij} = \frac{X_{ij}}{\sqrt{\{x^T\}[M]\{x\}}} \quad (2.38a)$$

Dimana ϕ_{ij} adalah komponen I yang dinormalisasi dari modal vektor j. Untuk normalisasi eigenvector, kondisi orthogonal harus terpenuhi

$$\{\phi\}_i^T [M] \{\phi\}_j = I \rightarrow i \neq j \quad (2.38b)$$

$$\{\phi\}_i^T [M] \{\phi\}_j = 0 \rightarrow i = j \quad (2.38c)$$

Selain itu kondisi orthogonal didapatkan dengan persamaan undamped free vibration untuk mode j yang dinormalisasi sebagai

$$\{\phi\}_i^T [K] \{\phi\}_j = 0 \rightarrow i \neq j \quad (2.38d)$$

$$\{\phi\}_i^T [K] \{\phi\}_j \neq 0 \rightarrow i = j \quad (2.38e)$$

Relasi persamaan di atas serupa dengan

$$[\phi]^T [M] [\phi] = [I] \quad (2.38f)$$

dan

$$[\phi]^T [K] [\phi] = [\Omega] \quad (2.38g)$$

Pemilihan normalisasi tergantung kepada kebutuhan dan acuan yang digunakan dalam analisa dinamik.

2.2.2.4. Matriks Redaman Klasik untuk Analisa Dinamis

Salah satu tipe redaman paling umum yang digunakan pada analisa non linear model matriks redaman adalah yang diasumsikan proporsional terhadap matriks massa dan matriks kekakuan, akan dibentuk seperti persamaan dibawah ini:

$$C = a_0 \mathbf{M} + a_1 \mathbf{K} \quad (2.39)$$

Tipe ini seringkali dikenal dengan Redaman Rayleigh. Pada analisa modal, matriks redaman harus memenuhi properti untuk membentuk persamaan modal yang tidak saling terikat sebagaimana persamaan di bawah ini:

$$2\zeta_n \omega_n = \phi_n^T C \phi_n = a_0 \phi_n^T \mathbf{M} \phi_n + a_1 \phi_n^T \mathbf{K} \phi_n \quad (2.40a)$$

$$0 = \phi_n^T C \phi_m \quad n \neq m \quad (2.40b)$$

Karena terdapat syarat orthogonalitas dari properti massa dan kekakuan, maka persamaan di atas dapat dituliskan dengan:

$$2\zeta_n \omega_n = a_0 + \delta \omega_n^2 \text{ atau } \zeta_n = \frac{1}{2\omega_n} a_0 + \frac{\omega_n}{2} \delta \quad (2.41)$$

Hal ini menunjukkan bahwa redaman modal hanya dapat dispesifikan secara langsung pada dua frekuensi, i dan j, untuk menyelesaikan a_0 dan a_1 menggunakan persamaan di bawah ini:

$$\begin{bmatrix} \xi_i \\ \xi_j \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{1}{\omega_i} & \omega_i \\ \frac{1}{\omega_j} & \omega_j \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \text{ untuk } \xi_i = \xi_j = \xi \quad \begin{cases} a_1 = \frac{2\delta}{\omega_i + \omega_j} \\ a_0 = \omega_i \omega_j \delta \end{cases} \quad (2.42)$$

Untuk kasus khusus, redaman diatur memiliki nilai yang sama dengan kedua frekuensi, oleh karena itu $\xi_i = \xi_j = \xi$ dan faktor proporsional dihitung dengan:

$$a_1 = \frac{2\delta}{\omega_i + \omega_j} \text{ dan } a_0 = \omega_i \omega_j \delta \quad (2.43)$$

Asumsi massa proporsional terhadap redaman menunjukkan pengaruh peredam dukungan yang berasal dari luar yang secara fisik tidak mungkin untuk penopang pada bawah struktur, sedangkan kekakuan yang proporsional terhadap redaman memberikan pengaruh peningkatan redaman pada mode yang lebih tinggi pada struktur yang secara fisik tidak dapat dipastikan pengaruhnya. Bentuk dari redaman ini menghasilkan kesalahan yang signifikan berdasarkan masalah tipe penyebabnya dan perpindahan gempa sebagai input

pada bawah struktur. Oleh karena itu, penggunaan redaman *rayleigh-type* ini sulit untuk untuk dipastikan sesuai pada kebanyakan struktur. Bagaimanapun, hal ini akan dilanjutkan dengan bantuan komputer untuk mengeluarkan hasil numerik dengan integral time step dalam jangka waktu yang panjang.

2.2.2.5. Metode *Time History*

Penentuan respon struktur pembebanan dinamis khususnya akibat eksitasi pondasi terdapat beberapa cara. Pada penelitian ini digunakan metode analisa *time history* yang akan dijelaskan sebagai berikut:

Respon dari struktur sendiri dinyatakan dengan hubungan superposisi linier dari *mode shape* sebagaimana persamaan (2.9) maka kita membutuhkan nilai $y(t)$, pada metode *time history* nilai $y(t)$ yang dipecahkan dengan persamaan di bawah ini:

$$y(t) = \Gamma_i \int_0^t p(\tau) e^{\zeta \omega_n(t-\tau)} \sin[\omega_n(t-\tau)] d\tau \quad (2.44)$$

Karena persamaan $p(\tau)$ adalah fungsi yang cukup rumit maka evaluasi dari persamaan ini dengan metode numerik (Clough, 2003). Keteletian penyelesaian yang akan diperoleh dari setiap proses numerik tentu saja bergantung pada panjang interval waktu (Δt). Pada umumnya, interval ini harus cukup pendek untuk fungsi-fungsi trigonometrik yang ingin ditentukan dnegan baik; $\Delta t \leq T/10$ adalah aturan yang umum untuk menyatakan hasil yang cukup baik.

- Evaluasi Metode Numerik

Terdapat beberapa metode numerik yang telah digunakan untuk menentukan solusi persamaan gerak pada sebuah sistem seperti *Method Based on Interpolation of Excitation*, *Central Difference Method*, *Wilson* dan *Newmark's Method* (Anil Chopra, 2011). Teknik numerik di atas adalah teknik yang dapat digunakan pada sistem linear maupun non linear. Pemilihan metode di atas dipertimbangkan berdasarkan kecocokan sifat untuk keadaan deformasi sistem non-linear pada setiap pertambahan waktu.

Karena penelitian ini didasari sistem elastis maka pemilihan metode tidak terlalu tergantung pada kesesuaian dengan sifat-sifatnya, adapun pada penelitian ini digunakan metode *Newmark* untuk sistem derajat kebebasan banyak yang dijelaskan sebagai berikut:

Penyelesaian dengan *Newmark Method* terbagi menjadi dua, yaitu *constant average acceleration* ($\gamma=1/2$ dan $\beta=1/4$) dan *linear acceleration* ($\gamma=1/2$ dan $\beta=1/6$). Jelas

dinyatakan oleh kedua metoda tersebut, maka pada metoda pertama yaitu percepatan linier, percepatan yang digunakan terus berubah terhadap waktu selama selang waktunya sehingga membentuk sebuah grafik linier dan metode percepatan konstan rata-rata adalah percepatan yang dirata-ratakan, sehingga tidak ada perubahan percepatan di setiap waktunya.

Adapun langkah-langkah yang dilakukan adalah sebagai berikut:

1. Perhitungan awal

$$a. [q_n]_0 = \frac{\phi_n^T m q_1}{\phi_n^T m \phi_n} \quad [\dot{q}_n]_0 = \frac{\phi_n^T m \dot{u}_1}{\phi_n^T m \phi_n}$$

$$[q_0^T] = [((q_1)_1, \dots, (q_1)_0)] \quad [\dot{q}_0^T] = [((\dot{q}_1)_1, \dots, (\dot{q}_1)_0)]$$

$$b. [P_0] = [\phi^T][p_0]$$

$$c. P_0, q_0, \text{ dan } \Delta t \text{ pada rentang waktu tertentu}$$

$$d. \text{ Evaluasi nilai percepatan modal pada waktu } i$$

$$[\ddot{q}_0] = [M] \cdot [P_0 - C\dot{q}_0 - Kq_0]^{-1}$$

$$e. \text{ Masukkan nilai } a_1, a_2, \text{ dan } a_3$$

$$[a_1] = \frac{1}{\beta(\Delta t)^2} [M] + \frac{\gamma}{\beta\Delta t} [C]$$

$$[a_2] = \frac{1}{\beta\Delta t} [M] + \left(\frac{\gamma}{\beta} - 1 \right) [C]$$

$$f. [a_3] = \left(\frac{1}{2\beta} - 1 \right) [M] + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) [C]$$

$$g. \text{ Tentukan nilai } [\hat{K}] = [K] + [a_1]$$

2. Perhitungan untuk setiap langkah waktu, $i = 0, 1, 2, \dots$

$$a. \text{ Menentukan } [P_{i+1}] = [\phi^T][p_{i+1}] + [a_1][q_1] + [a_2][q_2] + [a_3][q_3]$$

$$b. \text{ Menentukan } [q_{i+1}] = [\hat{K}]^{-1} \cdot [P_{i+1}]$$

$$c. \text{ Menentukan } [\dot{q}_{i+1}] = \frac{\gamma}{\beta\Delta t} (q_{i+1} - q_i) + \left(1 - \frac{\gamma}{\beta} \right) [\dot{q}_i] + \Delta t \left(1 - \frac{\gamma}{2\beta} \right) [\ddot{q}_i]$$

$$d. \text{ Menentukan } [\ddot{q}_{i+1}] = \frac{1}{\beta(\Delta t)^2} (q_{i+1} - q_i) - \left(\frac{1}{\beta\Delta t} \right) [\dot{q}_i] - \Delta t \left(\frac{1}{2\beta} - 1 \right) [\ddot{q}_i]$$

$$e. \text{ Menentukan } [u_{i+1}] = [\phi] \cdot [q_{i+1}]$$

3. Pengulangan untuk langkah waktu berikutnya. Ganti i dengan $i+1$ dan ulangi langkah a, b dan c persamaan di atas untuk waktu berikutnya.

Kestabilan metode *Newmark* ini tergantung dari ukuran time step yang digunakan. Semakin panjang time step-nya semakin tidak stabil, tetapi jika terlalu kecil kadang cukup merugikan karena memakan waktu yang cukup lama dan biaya yang lebih besar.

Untuk percepatan jenis linear acceleration yang memiliki nilai $\gamma=1/2$ dan $\beta=1/6$, batas stabilnya

$$\frac{\Delta t}{T} < \frac{\sqrt{3}}{\pi} \quad (2.45)$$

Sedangkan untuk constant average acceleration yang memiliki nilai $\gamma=1/2$ dan $\beta=1/4$, batas stabilnya

$$\frac{\Delta t}{T} < \infty \quad (2.46)$$

yang menunjukkan bahwa metode *constant average acceleration* adalah tidak terikat dengan pemilihan time step atau tanpa syarat stabil. (Villaverde, 2009)

(Halaman ini sengaja dikosongkan)

BAB III

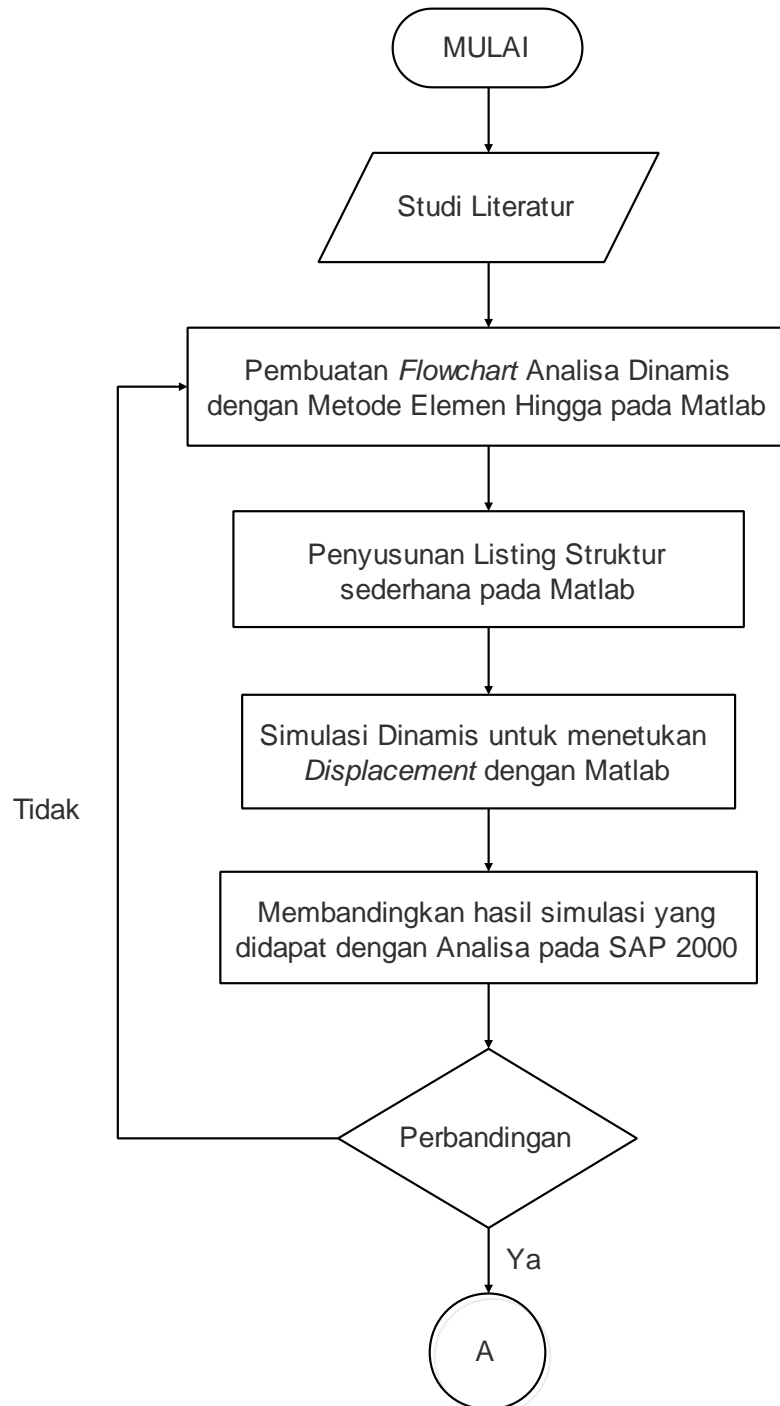
METODOLOGI PENELITIAN

BAB III

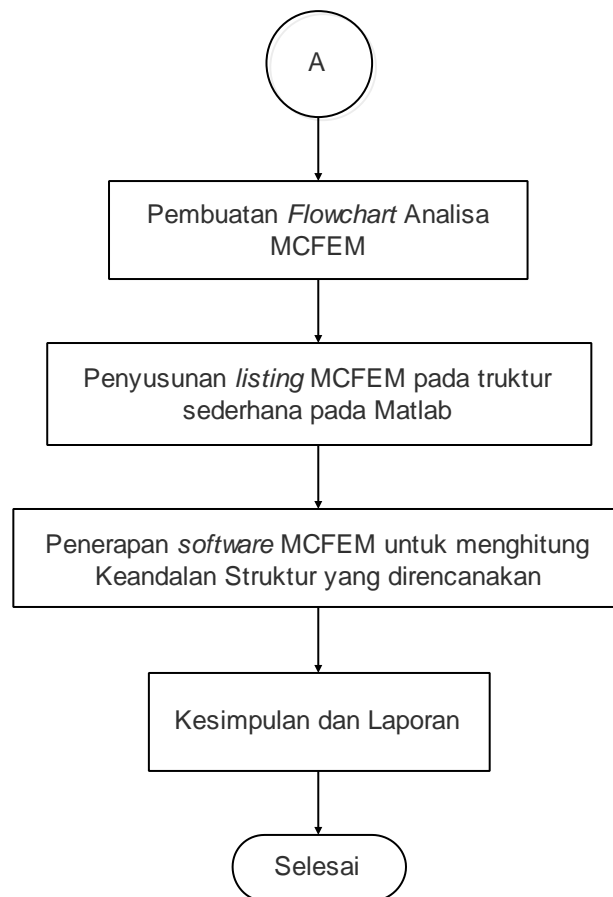
METODOLOGI PENELITIAN

3.1. Metodologi Penelitian

Penjelasan mengenai tugas akhir dapat dilihat pada Gambar 3.1 sebagai berikut:



Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir



Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir (Lanjutan)

3.2. Prosedur Penelitian

Adapun langkah – langkah penelitian berdasarkan pada diagram alir diatas dapat dijelaskan sebagai berikut:

a. Studi Literatur

Dilakukan proses mencari literatur yang relevan yang ditujukan untuk mendapatkan informasi tentang *Monte Carlo*, Metode Elemen Hingga pada Struktur Dinamis, penggunaan *software Matlab* untuk analisa elemen hingga dan lain sebagainya. Studi literatur ini dari sumber diantaranya adalah sebagai berikut : adalah buku, jurnal, penelitian sebelumnya termasuk tugas akhir terdahulu yang memiliki topik yang berkaitan.

b. Pembuatan *Flowchart* Metode Elemen Hingga

Dilakukan penyusunan *flowchart* algoritma perhitungan *Displacement* model dinamis struktur dengan Metode Elemen Hingga. *Flowchart* ini merupakan prosedur yang menjelaskan langkah pengerjaan dalam bentuk diagram alir.

- c. Penyusunan *Listing* Struktur sederhana
Pada tahapan ini, setelah menyusun *flowchart* perhitungan metode elemen hingga, akan dilakukan penyusunan *listing* fungsi pada *Matlab*. Dengan penyusunan *listing* ini, akan didapatkan software yang siap digunakan.
- d. Simulasi Analisa Dinamis
Selanjutnya dilakukan analisa dinamis akibat eksitasi pondasi. Analisa model dinamis didapatkan dari hasil *software* metode elemen hingga yang di buat pada *Matlab* sehingga di dapatkan *displacement*.
- e. Perbandingan Model
Setelah hasil *displacement* pada *Matlab* diketahui, kemudian dilakukan analisa yang sama pada *software* SAP2000 untuk dibandingkan hasilnya. Untuk hasil yang sama maka akan dilanjutkan ke langkah analisa keandalan namun jika hasil *Displacement* yang dikeluarkan berbeda maka dilakukan perbaikan pada algoritma pemrograman *Matlab*.
- f. Pembuatan *Flowchart Monte Carlo Finite Element Method*
Dilakukan penyusunan *flowchart* algoritma perhitungan keandalan model dinamis struktur sederhana dengan *Monte Carlo Finite Element Method*.
- g. Penyusunan *Listing* Struktur sederhana
Pada tahapan ini, setelah menyusun *flowchart* perhitungan keandalan struktur model dinamis, akan dilakukan penyusunan *listing* fungsi pada *Matlab*. Dengan penyusunan *listing* ini, akan didapatkan software yang siap digunakan.
- h. Perhitungan Keandalan Struktur
Pada tahap ini, *listing* algoritma analisa dinamis yang telah dibandingkan dengan software komersial selanjutnya akan diterapkan pada *Monte Carlo Finite Element Method* sehingga dapat diketahui keandalan struktur terhadap *Displacement* yang terjadi.
- i. Kesimpulan dan Laporan
Setelah analisa dan pembahasan dilakukan, maka dapat ditarik kesimpulan dari permasalahan dan tujuan sebelumnya, seberapa besar keandalan struktur dari analisa dinamis yang dilakukan.

(Halaman ini sengaja dikosongkan)

BAB IV

ANALISA DAN PEMBAHASAN

BAB IV

ANALISA DAN PEMBAHASAN

4.1. *Monte Carlo Finite Elemen Method* (MCFEM) pada Model Dinamis

Perhatian penting dalam analisa keandalan struktur adalah untuk menemukan respon beban. Untuk masalah rumit, seringkali kemungkinan yang terjadi diprediksi dengan analisa numeris untuk menghitung respon beban. *Finite Element* adalah salah satu solusi untuk memberikan hasil analisa numeris dengan bantuan Komputer.

Metode elemen hingga ini dibangun atas dasar variabel yang sudah ditentukan, tapi faktanya geometri dari ukuran struktur, beban, dan ketahanan material, dll, biasanya merupakan variabel random. Untuk mengatasi hal ini maka salah metode yang paling efektif adalah Metode *Monte Carlo*.

Adapun langkah-langkah penggunaan *Monte Carlo Finite Elemen Method* untuk mengidentifikasi keandalan suatu struktur model dinamis adalah sebagai berikut:

- a. Identifikasi distribusi dari semua parameter dasar yang berkenaan dengan keandalan struktur

Pertama kali yang dilakukan dalam menggunakan analisa MCFEM adalah menentukan parameter distribusi untuk bentuk, beban, dan ketahanan material, dll., berdasarkan keandalan struktur yang telah diberikan.

Berdasarkan hasil analisa statistik identifikasi distribusi parameter berdasarkan sebagai berikut :

- i. Distribusi bentuk struktur secara umum merupakan distribusi normal.
- ii. Asumsi atas nilai toleransi dan standar deviasi harus dikaitkan berdasarkan persamaan.

$$\sigma_x = \frac{\Delta X}{3} \quad (4.1)$$

Distribusi ini diperoleh hanya dari analisa statistik dari besaran data yang diperoleh dari struktur lapangan.

- iii. Distribusi yang kita gunakan nanti merupakan hasil dari uji laboratorium.

- b. Menghasilkan distribusi dari pemanggilan data acak

Setelah distribusi dari tiap variabel acak yang ditentukan, selanjutnya diperlukan adanya *random sampling* dari setiap variabel di atas. Untuk memperoleh *random sampling* dengan bentuk dan beban dengan cepat dan tepat, maka dilakukan dua

langkah berikut yaitu pertama memanggil data acak yang mempunyai distribusi seragam antara 0 – 1, kedua mentransformasi bilangan tersebut sehingga sesuai dengan variabel-variabel di atas. Pada penelitian yang dilakukan Guoliang ini, data acak yang diperlukan tersebut didapat dengan *multiplier common modulus method*. Metode ini, diekspresikan dalam Persamaan 4.2 dan 4.3 sebagai berikut:

$$X(i+1) = (\lambda * x(i)) * \text{mod} \quad (4.2)$$

$$r(i+1) = x(i+1) / m \quad (4.3)$$

dimana, λ dan m adalah bilangan integer positif.

Data acak yang dihasilkan dengan metode ini adalah *puppet random data*/data acak boneka. Jika nilai dari λ , $x(0)$ dan m adalah pilihan variabel yang masuk akal, maka *puppet random data* yang ada telah lulus dalam hal uji parameter, uji keseragaman distribusi dan uji independensi. Kemudian, rangkaian *puppet random data* dalam jumlah besar, keseragaman yang bagus dan murni berupa bilangan acak telah didapatkan.

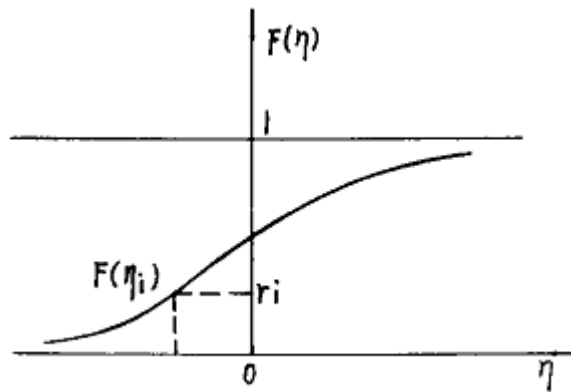
- c. Menentukan *Random Sample* dari distribusi yang didapatkan

Data acak antara 0 – 1 dibutuhkan sebagai macam distribusi acak yang memberikan berbagai variabel random. Disini kita membutuhkan input dari program elemen hingga.

Metode penentuan *random sample* yang banyak digunakan adalah *Inverse function Method*. Cara ini cocok untuk data diskrit maupun berupa distribusi menerus. Seandainya diduga variabel acak adalah η dan mempunyai fungsi distribusi $F(\eta)$, maka berdasarkan karakter dari fungsi distribusinya, dapat diketahui bahwa variabel selalu naik secara monoton dan dapat mempunyai nilai antara 0 – 1, kemudian kita mempunyai Persamaan 4.4 berikut :

$$r = F(\eta) = \int_{-\infty}^{\eta} f(\eta) d\eta \quad (4.4)$$

Ketika nilai dari η telah diperoleh antara $(-\infty, \eta_i)$, nilai r harus mempunyai nilai $r(i)$ antara $(0, F(\eta_i))$. Dimana $r(i)$ adalah nilai sembarang dari distribusi seragam antara $(0,1)$ (lihat Gambar 4.1). Dengan pengertian tersebut, maka dapat ditentukan nilai sampel dari berbagai distribusi variabel acak dengan memperhatikan transisi nilai distribusi acak yang seragam antara $(0,1)$.



Gambar 4.1. Diagram Penentuan Sampel Variabel Acak (Guoliang et. al., 1993)

d. Perhitungan elemen hingga berdasarkan *Random Sampling*

Setelah didapatkan *random sampling* pada ukuran struktur dan beban eksternal sebagaimana metode di atas, selanjutnya dari beberapa grup data sampel membentuk distribusi yang sama. Kemudian dari beberapa grup data sampel tersebut dimasukkan kedalam program metode elemen hingga, beberapa nilai tegangan atau defleksi pada titik-titik rawan pada struktur yang akan dianalisa untuk diketahui.

e. Menentukan Fungsi Moda Kegagalan, $MK(X)$

Fungsi moda kegagalan $MK(X)$ dari struktur dinyatakan berdasarkan kriteria desain *serviceability limit state (SLS)*, dimana *SLS* menyatakan kegagalan yang terjadi pada kondisi operasi normal disebabkan oleh limit dari defleksi atau getaran normal yang terjadi (Paik & Kumar, 2003).

Pada kenyataanya, deformasi berlebih pada struktur juga indikasi dari kelebihan getaran atau *noise*.

Tujuan utama dari *SLS* ini menjadikan struktur bekerja dengan efisien dan ekonomis tanpa dilakukan perawatan rutin. Batas yang diterima tergantung pada jenis, misi, dan susunan struktur. Lebih jauh, dalam mendefinisikan batas struktur, setiap disiplin berbeda seperti pada analisa keandalan struktur gempa, *SLS* memberikan kriteria bahwa stuktur tetap bertahan dengan intensitas kecil gempa tanpa mengalami kerusakan (Dymiotis et al, 2001). Untuk pada analisa struktur akibat beban gempa yang *displacement*-nya dibatasi dinyatakan dengan persamaan berikut:

$$\Delta u_e \leq [\theta_e] h \text{ atau } [\theta_e] h - \Delta u_e \geq 0 \quad (4.4)$$

Dimana persamaan di atas digunakan sebagai persamaan moda kegagalan $MK(X)$ yang dapat dituliskan sebagai berikut:

$$MK(\mathbf{X}) \leq 0 \quad (4.5a)$$

$$u_{control} - u_{actual} \leq 0 \quad (4.5b)$$

Δu_e adalah displacement maksimum dari setiap tingkatan struktur.

θ_e adalah limit untuk displacement rotasi lantai elastis, dimana ketentuan limit ini diadopsi dari tabel 4.1.

h adalah tinggi struktur

Tabel 4.1 Limit untuk *Displacement* Tingkatan Struktur Elastis (Code for Seismic Design of Building, 2010)

Tipe struktur	θ_e
Reinforced concrete frame	1/550
Reinforced concrete frame-seismic wall, slab column-seismic wall, and frame-core-cube	1/800
Reinforced concrete seismic wall, tube-in-tube	1/1000
Reinforced concrete frame-supported storey	1/1000
Multi-storey and tall steel structure	1/250

Displacement Control sendiri didefinisikan sebagai *Displacement* batas yang tidak boleh dilampaui oleh *Displacement* pada struktur, ditentukan melalui persamaan dibawah ini:

$$\sigma = \frac{F}{A_0} \quad (4.6a)$$

$$\varepsilon = \frac{\Delta L}{L_0} \quad (4.6b)$$

Stress dan *strain* dihubungkan dengan Hukum Hooke

$$E = \frac{\sigma}{\varepsilon} \quad (4.6c)$$

Memasukkan persamaan 4.5a dan 4.5b kedalam persamaan hukum hooke

$$E = \frac{F / A_0}{\Delta L / L_0} \quad (4.6d)$$

atau dapat dituliskan dengan

$$\frac{\Delta L}{L_0} = \frac{\sigma}{E} \quad (4.6e)$$

Tentukan input dari masing-masing variabel, yaitu modulus young (E), panjang awal struktur (L_0), dan Stress (σ). Pada persamaan di atas stress diambil dari yield stress

struktur yang digunakan. Sehingga persamaan dapat dituliskan sebagai berikut untuk merepresentasikan *displacement control* (u_c) sebagai ganti dari Δt :

$$u_c = \frac{\sigma_y}{E} L_0 \quad (4.7)$$

Setelah variabel dimasukkan berupa angka acak dari distribusi tertentu, maka simulasi dilakukan secara berulang sebanyak iterasi yang didefinisikan dengan melakukan pemanggilan ke 3 parameter acak dengan bantuan *RNG* pada komputer yang disebutkan sebelumnya. Kemudian, ke-3 parameter ini ditransformasi ke variabel E , M_{add} , dan u_c baru yang digunakan sebagai input dalam perhitungan analisa dinamis untuk struktur *Plane Truss sederhana*.

Setelah didapatkan hasil berupa *Displacement* yang terjadi pada masing-masing *joint*, maka akan diidentifikasi apakah batang tersebut gagal atau tidak. Identifikasi akan dilakukan dengan menggunakan metoda moda kegagalan pada persamaan 4.5a.

Setiap *displacement* (u_{actual}) yang terjadi pada batang bernilai lebih besar dari *displacement control* ($u_{control}$) pada batang, maka struktur dinyatakan gagal dan dicatat 1 kali struktur gagal. Setelah mendapatkan data apakah struktur tersebut gagal atau sukses, maka percobaan diulangi kembali dengan memanggil *random number*. Begitu seterusnya hingga n kali simulasi dengan pencatatan beberapa kali struktur gagal. Setelah n kali simulasi, maka dapat ditentukan P_f (*probability of failure*) yaitu:

$$P_f = \frac{\text{JumlahKegagalan}(g)}{\text{TotalSimulasi}(n)} \quad (4.8)$$

Dari P_f yang didapatkan, maka dapat ditentukan keandalan masing-masing elemen maupun keandalan sistem yaitu:

$$R = 1 - P_f \quad (4.9)$$

Sebagaimana yang disinggung diawal bahwa keandalan dari sistem ini didapatkan dengan menerapkan monte carlo pada penyelesaian persamaan model dinamis, maka dalam proses menentukan keandalan setelah melalui beberapa analisa. Diataranya dilakukan analisa modal untuk mendapatkan matriks modal dari konfigurasi struktur berdasarkan elemen yang dibentuk. Setelah mendapatkan matriks modal, maka dilanjutkan dengan analisa dinamis.

Untuk menyelesaikan persamaan gerak model dinamis ini digunakan teknik numerik. Metode *Newmark* adalah metode yang digunakan dalam teknik numerik pada penelitian ini.

Ada beberapa cara dalam penyelesaian dengan teknik numerik ini, biasanya pemilihan dilakukan berdasarkan kesesuaian *output* dari grafik. Selain itu syarat baiknya grafik *output* pada penyelesaian teknik numerik ini harus akurat, konvergen, stabil dan dapat diaplikasikan pada komputer.

Adapun metode *Newmark* adalah metode bertahap (*time-stepping method*) yang mempunyai persamaan dasar sebagai berikut:

$$[P_{i+1}] = [\phi^T][P_{i+1}] + [a_1][q_1] + [a_2][q_2] + [a_3][q_3] \text{ an} \quad (4.10)$$

Dimana setiap variabel a_1 , a_2 , dan a_3 ini memiliki variabel γ dan β didalamnya. Nilai parameter γ dan β tergantung pada percepatan apa yang diasumsikan pada metode *Newmark* yang digunakan.

Simulasi yang disarankan mengacu kepada *Probability of Failure* (P_f) DnV RP G101 dimana skala yang memungkinkan dilakukan pada batas kegagalan tidak terjadi pada 10^{-5} selain itu menurut Wisudawan (2013) simulasi ini akan dihentikan jika memenuhi syarat konvergensi (ϵ) yang ditetapkan yaitu pada persamaan 4.11a dan 4.11b di bawah ini:

$$\frac{(u_{xi} + 1) - u_{xi}}{u_{xi}} \times 100\% \leq 0.006\% \quad (4.11a)$$

$$\frac{(R_i + 1) - R_i}{R_i} \times 100\% \leq 0.6\% \quad (4.11b)$$

Spesifikasi komputer yang digunakan akan mempengaruhi lama waktu *running* saat simulasi, adapun simulasi pada penelitian ini akan dilakukan dengan komputer yang mempunyai spesifikasi dasar sebagai berikut:

- ✓ Processor = Intel ® Core™ i5 CPU @ 3.20 GHz
- ✓ RAM = 4,00 GB
- ✓ Sistem Operasi = Windows 8, 64-bit

4.2. Algoritma dan listing fungsi Aplikasi Simulasi Monte Carlo pada Struktur Model Dinamis

Berdasarkan hasil studi literatur dan telaah pustaka berkaitan dengan analisa struktur model dinamis, maka diperlukan beberapa tahap dalam menentukan perpindahan pada

struktur model dinamis yaitu menentukan konfigurasi struktur berdasarkan jenis elemen yang diinginkan, melakukan analisa modal dan menentukan metode yang digunakan dalam menentukan perpindahan yang terjadi.

Adapaun algoritma penyelesaian persamaan gerak model dinamis dengan menggunakan teknik numerik akan dijelaskan langkah-langkah pengerjaannya pada gambar 4.2.

Algoritma pada gambar 4.2 dapat dijealskan sebagai berikut:

- a. Tentukan banyaknya simulasi yang akan kita lakukan.
- b. Input *random variabel* yang telah ditentukan.
- c. Matriks kekakuan elemen truss global dan matriks massa massa terkumpul global dapat dituliskan sebagai berikut:

$$\hat{k} = \frac{AE}{L} \begin{bmatrix} C^2 & SC & -C^2 & -SC \\ & S^2 & -SC & -S^2 \\ \text{Simetri} & & C^2 & SC \\ & & & S^2 \end{bmatrix} \quad (4.12)$$

dan

$$\hat{m} = \frac{\rho AL}{2} \begin{bmatrix} C^2 + S^2 & SC - SC & 0 & 0 \\ SC - SC & C^2 + S^2 & 0 & 0 \\ 0 & 0 & C^2 + S^2 & SC - SC \\ 0 & 0 & SC - SC & C^2 + S^2 \end{bmatrix} \quad (4.13)$$

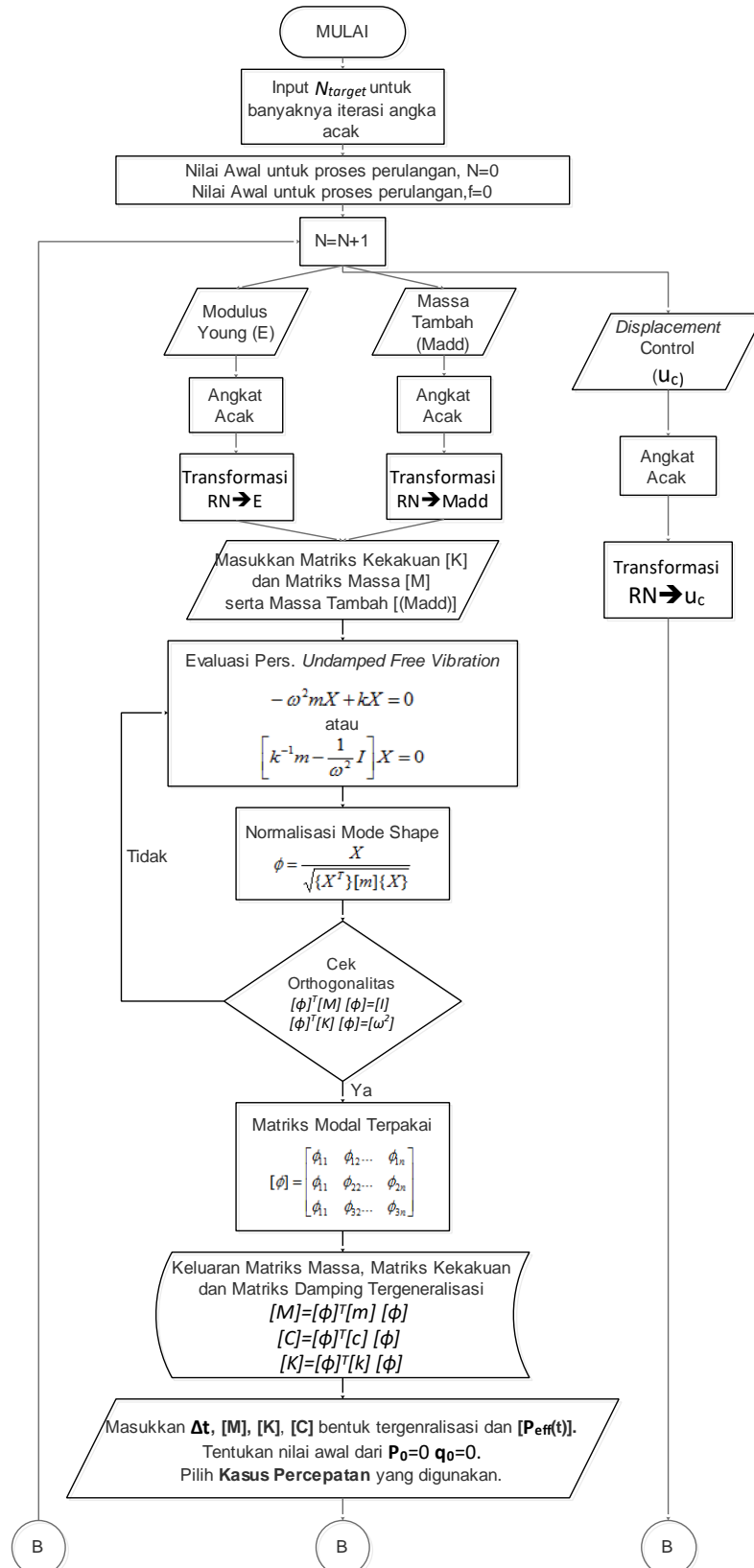
Matriks massa dan kekauan di atas akan digunakan sebagai input pada analisa modal.

- d. Untuk mendapatkan nilai dari frekuensi natural dan moda getar dilakukan dengan meyelesaikan persamaan berikut:

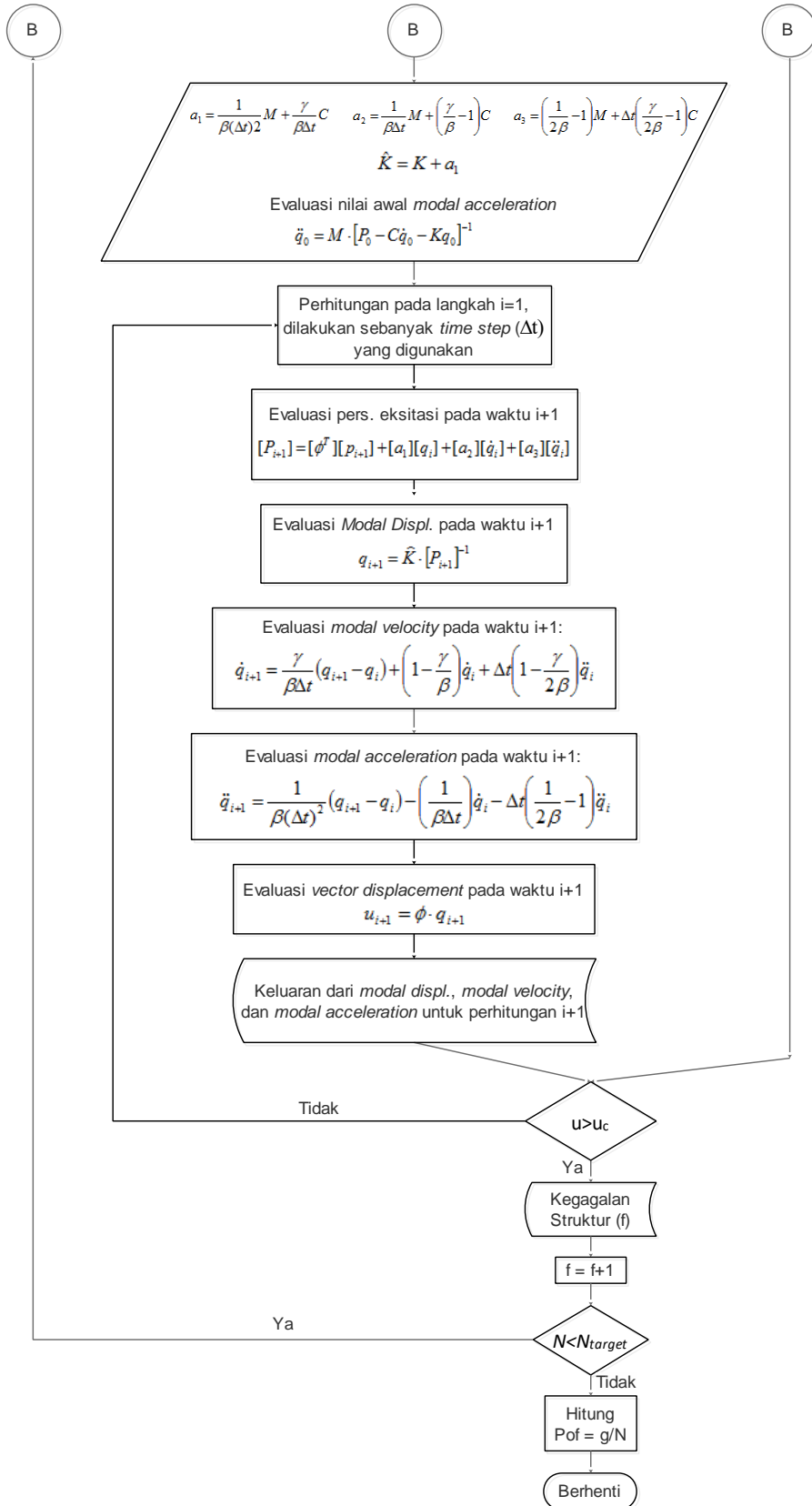
$$k^{-1}mX = \frac{1}{\omega^2}IX \quad (4.14a)$$

atau dapat dituliskan

$$[A - \lambda I]X = 0 \quad (4.14b)$$



Gambar 4.2 Algoritma Solusi Persamaan Gerak Struktur Model Dinamis



Gambar 4.2 Algoritma Solusi Persamaan Gerak Struktur Model Dinamis (Lanjutan)

- e. Moda getar perlu dilakukan cek orthogonalitas dengan menggunakan persamaan di bawah ini:

$$\phi_{ij} = \frac{x_{ij}}{\sqrt{\{x^T\}[M]\{x\}}} \quad (4.15a)$$

Dimana ϕ_{ij} adalah komponen I yang dinormalisasi dari modal vektor j. Untuk normalisasi eigenvector, kondisi orthogonal harus terpenuhi

$$[\phi]^T [M] [\phi] = [I] \quad (4.15b)$$

dan

$$[\phi]^T [K] [\phi] = [\Omega] \quad (4.15c)$$

- f. Pilih kejadian percepatan yang terjadi pada struktur diantaranya adalah sebagai berikut:

- i. Metode Percepatan Rata-Rata (*average acceleration*)

Metode ini diasumsikan percepatan yang terjadi adalah percepatan yang telah dirata-ratakan. Sehingga tidak ada perubahan percepatan disetiap waktunya. Memiliki nilai $\gamma=1/2$ dan $\beta=1/4$.

- ii. Metode Percepatan Linier (*linear acceleration*)

Pada metode percepatan linear ini, percepatan yang digunakan berubah berdasarkan waktu. Sehingga membentuk grafik yang linear dengan nilai $\gamma=1/2$ dan $\beta=1/6$.

- g. Masukkan matriks kekakuan $[K]$, massa $[M]$, redaman $[C]$, gaya $[P_{eff}(t)]$ bentuk tergeneralisasi. Masukkan juga kondisi awal untuk $t=0$ yang dinyatakan sebagai berikut:

- i. P_0 , q_0 , dan Δt pada rentang waktu tertentu sebagai bentuk kondisi awalnya.

- ii. Evaluasi nilai percepatan modal pada waktu i

$$[\ddot{q}_0] = [M] \cdot [P_0 - C\dot{q}_0 - Kq_0]^{-1}$$

- iii. Masukkan nilai a_1 , a_2 , dan a_3

$$[a_1] = \frac{1}{\beta(\Delta t)^2} [M] + \frac{\gamma}{\beta\Delta t} [C]$$

$$[a_2] = \frac{1}{\beta\Delta t} [M] + \left(\frac{\gamma}{\beta} - 1 \right) [C]$$

$$[a_3] = \left(\frac{1}{2\beta} - 1 \right) [M] + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) [C]$$

- iv. Tentukan nilai $[\hat{K}] = [K] + [a_1]$
- h. Melakukan perhitungan untuk setiap langkah waktu, $i = 0, 1, 2, \dots$
 - i. Menentukan $[P_{i+1}] = [\phi^T][p_{i+1}] + [a_1][q_1] + [a_2][q_2] + [a_3][q_3]$
 - ii. Menentukan $[q_{i+1}] = [\hat{K}]^{-1} \cdot [P_{i+1}]$
 - iii. Menentukan $[\dot{q}_{i+1}] = \frac{\gamma}{\beta \Delta t} (q_{i+1} - q_i) + \left(1 - \frac{\gamma}{\beta}\right) [\dot{q}_i] + \Delta t \left(1 - \frac{\gamma}{2\beta}\right) [\ddot{q}_i]$
 - iv. Menentukan $[\ddot{q}_{i+1}] = \frac{1}{\beta (\Delta t)^2} (q_{i+1} - q_i) - \left(\frac{1}{\beta \Delta t}\right) [\dot{q}_i] - \Delta t \left(\frac{1}{2\beta} - 1\right) [\ddot{q}_i]$
 - v. Menentukan $[u_{i+1}] = [\phi] \cdot [q_{i+1}]$
- i. Cek *Displacement* yang melampaui *Displacement control*. Jika melampaui, maka akan dicatat sebagai sebuah kegagalan meskipun dalam proses iterasi Δt baik banyak atau sedikit tetap akan tercatat sebagai 1 kali kegagalan yang akan dihitung pada *probability of failure* dari total percobaan. Kemudian ulangi tahap *time step* ini. Ganti i dengan $i+1$ dan implementasikan langkah i, ii, dan iii pada poin g di atas untuk *time step* selanjutnya.
- j. Setelah melalui proses *time step* 1 kali, setiap kegagalan banyak atau sedikit tetap akan dihitung sebagai 1 kali proses bukan proses sebanyak *time step* yang dilakukan. Maka pada setiap kegagalan yang terjadi akan dituliskan sebagai data gagal.
- k. Iterasi percobaan simulasi monte carlo di ulangi lagi sampai memenuhi jumlah simulasi yang telah ditentukan diawal.
- l. Dilakukan perhitungan *probability of failure*.

Adapun listing dari pemrograman dari algoritma di atas akan dijelaskan sebagai berikut:

a. Listing Fungsi Kekakuan Elemen *Truss*

Listing fungsi ini digunakan untuk menentukan matriks kekakuan global elemen *Truss* pada masing-masing batang.

```
function y = PlaneTrussElementStiffness(E,A,L, theta)
%PlaneTrussElementStiffness This function returns the element
% stiffness matrix for a plane truss
% element with modulus of elasticity E,
% cross-sectional area A, length L, and
% angle theta (in degrees).
% The size of the element stiffness
% matrix is 4 x 4.
```

```

x = theta;
C = cosd(x);
S = sind(x);
y = E*A/L*[C*C C*S -C*C -C*S ; C*S S*S -C*S -S*S ;
           -C*C -C*S C*C C*S ; -C*S -S*S C*S S*S];

```

Input yang diperlukan dalam fungsi ini adalah *modulus young*, luas penampang batang, panjang batang, dan sudut inklinasi batang.

b. Listing Fungsi Massa Elemen Truss

Pada penelitian ini matriks massa diasumsikan sebagai *lumped parameter matrix* (massa terpusat pada setiap nodal), dimana listing fungsinya dinyatakan sebagai berikut:

```

function y = PlaneTrussElementMass(rho,A,L, theta)
%PlaneTrussElementStiffness This function returns the mass
% matrix for a plane truss
% element with mass density rho,
% cross-sectional area A, length L, and
% angle theta (in degrees).
% The size of the element stiffness
% matrix is 4 x 4.
x = theta;
C = cosd(x);
S = sind(x);

%for lumped mass use the following
y=(rho*A*L/2)*[(C*C)+(S*S) (S*C)-(C*S) 0 0;
                (C*S)-(S*C) (C*C)+(S*S) 0 0;
                0 0 (C*C)+(S*S) (S*C)-(C*S);
                0 0 (C*S)-(S*C) (C*C)+(S*S)];

```

Input yang diperlukan dalam fungsi ini adalah massa jenis, luas penampang dan panjang batang.

c. Listing Fungsi Matriks Gabungan

Listing fungsi matriks gabungan (*assembly matrix*) ini digunakan untuk superposisikan seluruh batang matriks kekakuan dan matriks massa struktur menjadi satu matriks kekakuan dan satu matriks massa saja. Listing fungsi dari matriks gabungan adalah sebagai berikut:

```

function y = PlaneTrussAssemble(K,k,i,j)
%PlaneTrussAssemble This function assembles the element stiffness
% matrix k of the plane truss element with nodes
% i and j into the global stiffness matrix K.

```

```

% This function returns the global stiffness
% matrix K after the element stiffness matrix
% k is assembled.
lm(1)=2*i-1;
lm(2)=2*i;
lm(3)=2*j-1;
lm(4)=2*j;

for m=1:4
    ii=lm(m);
    for n=1:4
        jj=lm(n);
        K(ii,jj)=K(ii,jj)+k(m,n);
    end
end
y = K;

```

Input dari fungsi di atas adalah matriks nol yang berukuran menyesuaikan dengan jumlah nodal, matriks kekakuan dan matriks massa masing-masing batang dan nodal yang membentuk batang (i dan j).

d. Listing Fungsi Analisa Modal

Listing fungsi ini digunakan untuk menentukan frekuensi natural dan moda getar dari model.

```

% Modal Analysis
[V,D] = eig(K,M) ;
[W,k] = sort(diag(D)) ;
V = V(:,k) ;

%Normalize Mode Shape
Factor = diag(V'*M*V) ;

Phi = V*inv(sqrt(diag(Factor))) ; %Mode Shape Sebenarnya
Omega = diag(sqrt(Phi'*K*Phi)) %Frekuensi Natural

% selecting only first two Natural Frequencies and Mode shapes
lambda = 2 ;
w = diag(Omega(1:lambda)) ;
phi = Phi(:,1:lambda) ;

% Reducing the Degree's of Freedom of M, K and P
M = phi'*M*phi ;
C = phi'*C*phi ;
K = phi'*K*phi ;

```

Input dari listing ini adalah matriks massa dan kekakuan yang sudah dalam bentuk matriks gabungan dan kondisi batas.

e. Pemilihan jenis percepatan

Listing fungsi ini digunakan untuk menentukan jenis percepatan yang diinginkan berdasarkan kesesuaian data.

```
%Pemilihan Jenis Percepatan
acceleration = 'Average' ;

switch acceleration
    case 'Average'
        gaama = 1/2 ;beta = 1/4 ;
    case 'Linear'
        gaama = 1/2 ;beta = 1/6 ;
end
```

Input yang diperlukan dalam fungsi ini adalah dengan memilih “average” atau “linear” pada baris bertuliskan “acceleration”.

f. Listing Nilai Awal

Listing fungsi ini digunakan untuk menentukan nilai awal yang akan dimasukkan pada proses *time step*.

Pada proses ini terdapat beberapa listing diataranya adalah sebagai berikut:

i. Penentuan *time step* yang nantinya digunakan seperti rentang waktu yang terjadi dan selisih waktu.

```
% Time step
ti = 0. ;
tf = 8. ;
dt = 0.004 ;
t = ti:dt:tf ;
nt = fix((tf-ti)/dt) ;
n = length(M) ;
```

ii. Penentuan kondisi awal pada nilai q_0 , \dot{q}_0 , \ddot{q}_0 , U_0 , dan P_0

```
% Initial Conditions
depl(:,1) = zeros ;
vel(:,1) = zeros ;
U(:,1) = phi*depl(:,1) ;
P0 = zeros(n,1) ;
accl(:,1) = M\ (P0-C*vel(:,1)-K*depl(:,1)) ; %1.3
```

Input dari listing ini sebgain besar merupakan matriks yang bernilai nol.

iii. Penentuan \hat{K} , konstanta γ dan konstanta β yang mengikuti jenis percepatan yang ditentukan diawal.

```

% Constants used in Newmark's integration
a1 = gaama/(beta*dt) ; a2 = 1/(beta*dt^2) ; %a1
a3 = 1/(beta*dt) ; a4 = (gaama/beta)-1 ; %a2
a5 = (1/(2*beta))-1 ; a6 = dt*((gaama/(2*beta))-1) ; %a3

a = a1*C+a2*M ; %a1
b = a3*M+a4*C ; %a2
c = a5*M+a6*C ; %a3
Kcap = K+a ; %1.6

```

Adapun input listing ini terintegrasi dengan pemilihan jenis percepatan yang diinginkan oleh pengguna.

g. Tahap Metode Bertahap (*time step*)

Listing fungsi ini digunakan untuk menentukan perpindahan vektor $\mathbf{u}(t)$ yang terjadi pada struktur. Sama pada penentuan kondisi awal, *listing* ini terdiri dari beberapa proses yang masih berkaitan dengan listing kondisi awal, adapun bentuk listingnya digambarkan sebagai berikut:

```

% Time step starts
for i = 1:nt
    Pk = P(1:lambda,i+1) ; %2.1
    delP = Pk+a*depl(:,i)+b*vel(:,i)+c*accl(:,i) ; %2.1
    depl(:,i+1) = Kcap\delP ; %2.2
    vel(:,i+1) = a1*(depl(:,i+1)-depl(:,i))-a4*vel(:,i)-
a6*accl(:,i) ; %2.3
    accl(:,i+1) = a2*(depl(:,i+1)-depl(:,i))-a3*vel(:,i)-
a5*accl(:,i) ; %2.4
    U(:,i+1) = phi*depl(:,i+1) ; %2.5
    u1 = abs(U(1,i+1)) ;
    u2 = abs(U(2,i+1)) ;
    if u1 >= Ucontrol
        hitung=hitung+1 ;
    end
    d02x = d02x+u1 ;
    d02z = d02z++u2 ;
    xc=xc+1 ;
end

```

Input yang diperlukan dalam fungsi ini pada perhitungan pertama kali akan melakukan input dari nilai kondisi awal, setelah melakukan proses perhitungan awal pada saat $i=1$ sudah sampai pada proses nilai vektor perpindahan $\mathbf{u}(t)$ didapatkan, maka dilanjutkan untuk $i=2, 3, 4, \dots, n$ untuk inputan yang digunakan adalah hasil dari nilai output $i=2-1, 3-1, 4-1, \dots, n-1$.

h. Adapun listing fungsi yang digunakan untuk melengkapi penyelesaian perasmaan model dinamis yaitu berupa listing keandalan. Untuk menghindari *out of memory*

maka perhitungan keandalan akan di batasi pada joint kritis. Adapun listing fungsi yang digunakan adalah sebagai berikut:

```
x = 100 ; % Banyaknya Simulasi yang dilakukan
hitung = 0 ;
simulasigagal = 0 ;
d02x = 0 ;
d02z = 0 ;

for i=1:x

    .
    .
    .
    % Baja A36
    e = abs(normrnd(29.2*10^6,0.075)) ;

    .
    .
    .
    % Global matrix calculation of element
    k1 = PlaneTrussElementStiffness(e,a,l1,90) ;
    k2 = PlaneTrussElementStiffness(e,a,l2,315) ;

    .
    .
    .
    Madd = abs(normrnd(57320.188,7451.62444)) ;

    % Apply boundary conditions for stiffness matrix and mass
    K = K(3:4,3:4) ;
    M = M(3:4,3:4)+diag([Madd;Madd]) ;

    % Time step starts
    for i = 1:nt
        .
        .
        .
        u1 = abs(U(1,i+1)) ;
        u2 = abs(U(2,i+1)) ;
        if u1 >= Ucontrol
            hitung=hitung+1 ;
        end
    end

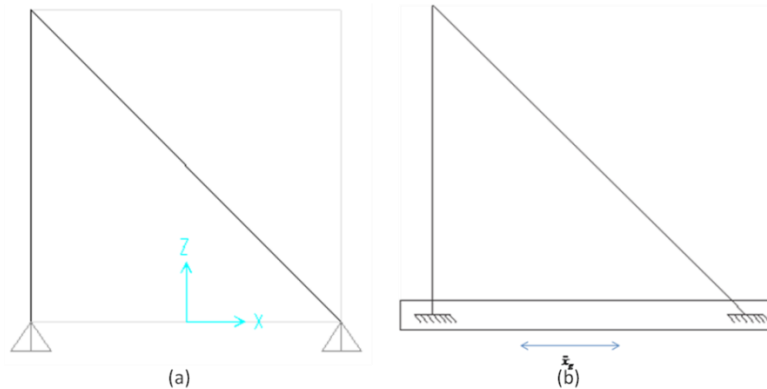
    struktur_gagal = hitung/nt ;

    if struktur_gagal > 0
        simulasigagal = simulasigagal+1 ;
    end
    d02x = d02x+u1 ;
    d02z = d02z+u2 ;
end
```

4.3. Hasil Perbandingan *Software*

Perbandingan *software* dilakukan untuk mengetahui seberapa akurat perhitungan dari simulasi yang dilakukan dengan membandingkan antara *Matlab* versi 2011 dengan *software SAP 2000* versi 14.

4.3.1. *Plane Truss* bentuk segitiga



Gambar 4.3 (a) Ilustrasi Model *Plane Truss* dan (b) Eksitasi pada pondasi *Plane Truss*

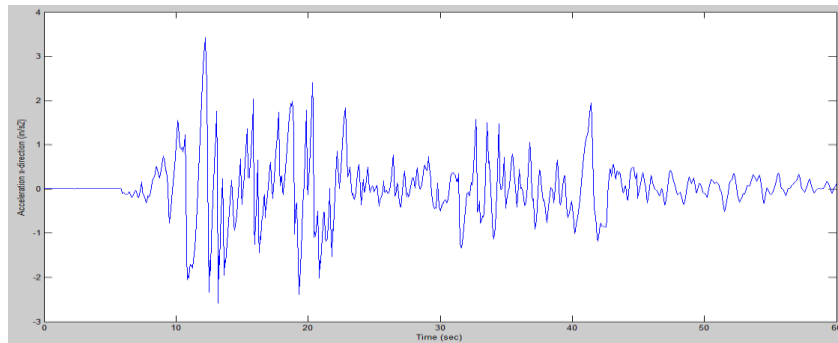
Model struktur yang dibentuk adalah elemen *truss* 2D sederhana berbentuk segitiga sebagaimana pada gambar di atas.

Perbandingan akan dilakukan pada struktur sederhana dengan *material properties* sebagai berikut:

Tabel 4.2 Material Properties Struktur Truss

Variabel	Harga
Modulus Young (E)	29000000 lb/in ²
Diameter (D)	12.3 in
Panjang Batang Horizontal (L1)	118.11 in
Panjang Batang Diagonal (L2)	167.0328 in
Massa jenis (ρ)	0.7 lb/in ³
Massa Tambahan (M_{add})	100000 lb

Percepatan yang terjadi pada tumpuan diambil dari percepatan gempa yang terekam ditunjukkan pada gambar di bawah ini:



Gambar 4.4 Percepatan yang terjadi pada Tumpuan Struktur

Adapun hasil analisa model dinamis pada struktur di atas dibandingkan dengan menggunakan *software SAP 2000*, hasil dari perbandingan dapat dilihat pada tabel 4.3, tabel 4.4 dan gambar 4.4 di bawah ini:

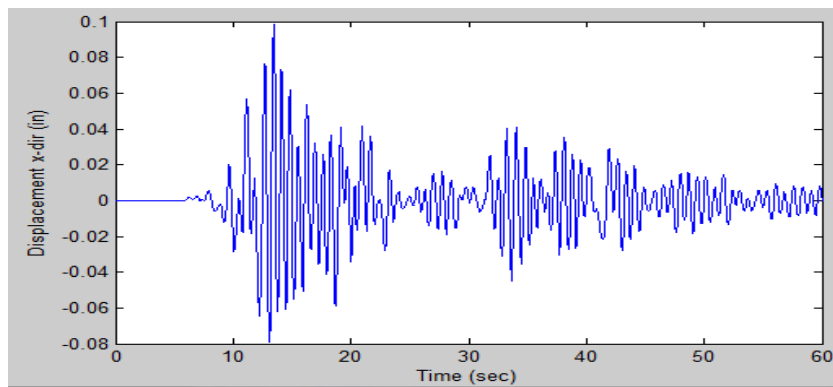
Tabel 4.3 Perbandingan Frekuensi Natural Model pada Matlab dan SAP 2000

	Mode	Matlab	SAP 2000	Selisih (%)	Unit
Frekuensi Natural	1	9.001500	9.001500	0.000002%	rad/s
	2	22.192151	22.192000	0.000681%	

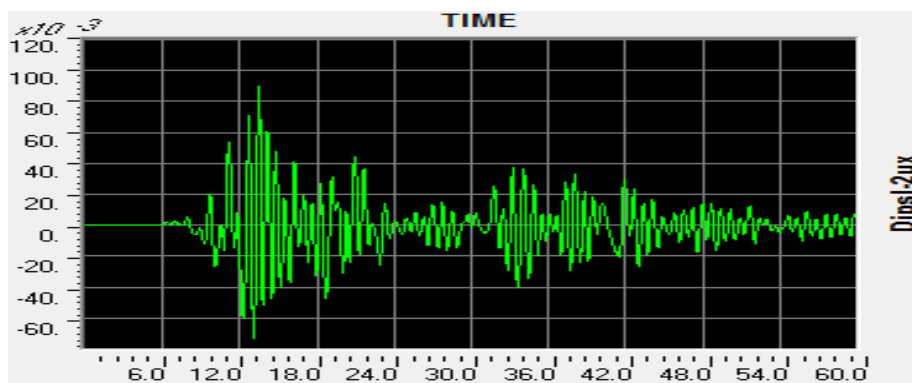
Tabel 4.4 Perbandingan *Displacement* Maksimum dan Minimum pada Matlab dan SAP 2000

		Proposed Software		Commercial Software		D (%)	
		(in)		(in)			
		U _x	U _z	U _x	U _z		
Displ.	Max	9.86E-02	3.22E-02	9.01E-02	2.95E-02	8.532%	8.380%
	Min	-7.90E-02	-2.28E-02	-7.31E-02	-2.13E-02	7.453%	6.372%

Displacement dalam bentuk grafik dinyatakan pada gambar di bawah ini:



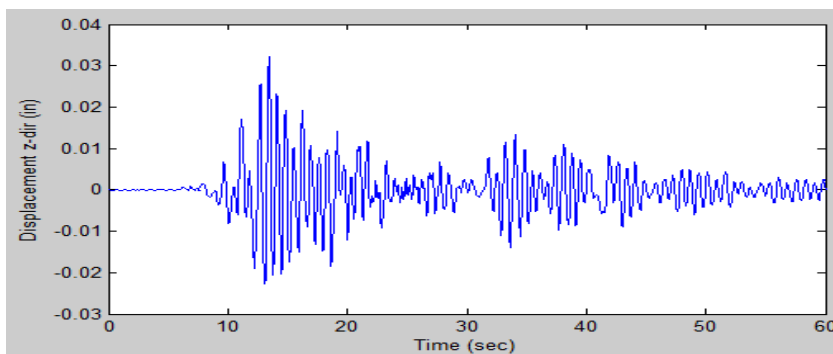
(a)



(b)

Gambar 4.5 Perbandingan Grafik *Displacement* pada Node 2 Sumbu-x (a) Matlab dan (b) SAP

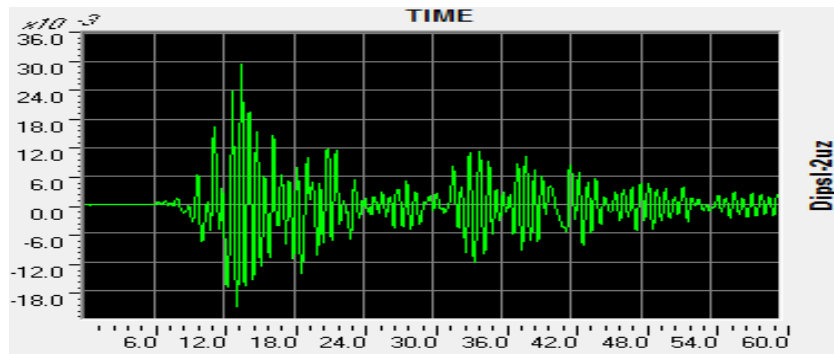
2000



(a)

Gambar 4.6 Perbandingan Grafik *Displacement* pada Node 2 Sumbu-z (a) Matlab dan (b) SAP

2000

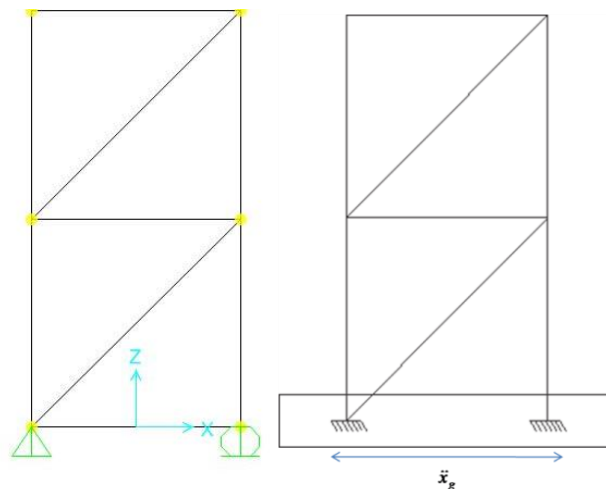


(b)

Gambar 4.6 Perbandingan Grafik *Displacement* pada Node 2 Sumbu-z (a) Matlab dan (b) SAP 2000 (Lanjutan)

Hasil perbandingan di atas, menunjukkan bahwa perhitungan *software* dengan menggunakan metode elemen hingga pada model dinamis mendekati hasil perhitungan pada *software* komersial atau SAP 2000 dan nilai *error* yang ditunjukkan oleh tabel terbilang kecil.

4.3.2. Struktur *Truss* 2 Tingkat



Gambar 4.7 Model Elemen *Truss* 2D Hasil simulasi

Perbandingan yang kedua, akan dilakukan pada struktur *Plane Truss* 2 tingkat dengan model percepatan pada tumpuan sebagaimana gambar di atas. Untuk perbandingan pada struktur ini juga sama seperti pada model pertama. Struktur ini memiliki *material properties* sebagai berikut:

Tabel 4.5 Properti Material Struktur *Truss* 2 Tingkat

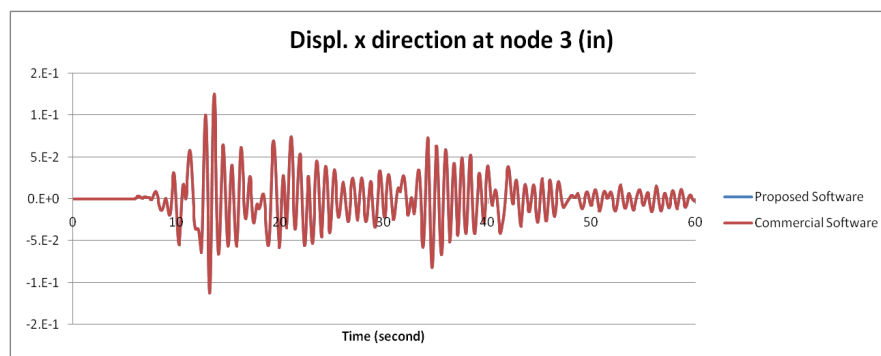
Variabel	Harga
Modulus Young (E)	29000000 lb/in ²
Diameter (D)	28.2 in
Panjang Batang Horizontal (L1)	118.11 in
Panjang Batang Diagonal (L2)	167.0328 in
Massa jenis (ρ)	0.7 lb/in ³
Massa Tambahan (M_{add})	100000 lb

Data dari tabel di atas akan dimasukkann ke dalam *listing* fungsi yang telah dibuat pada Matlab dan SAP 2000, sehingga akan memperlihatkan perbandingan hasil frekuensi natural pada tabel 4.6 berikut:

Tabel 4.6 Perbandingan Hasil Frekuensi Natural Struktur *Plane Truss* 2 Tingkat

	Mode	Matlab	SAP 2000	Δ
		(rad/s)		(%)
Natural Frequency	1	7.58433	7.5843	0.000%
	2	27.6778	27.678	0.001%
	3	35.7385	35.738	0.001%
	4	56.5218	56.522	0.000%
	5	613.128	613.15	0.004%
	6	1601.59	1601.6	0.000%
	7	1682.35	1682.4	0.003%
	8	1759.79	1759.8	0.000%
	9	1863.25	1863.3	0.003%

Perbandingan juga ditunjukkan dengan hasil perhitungan *Displacement* pada tabel 4.6 di bawah ini. Detail perhitungan dapat dilihat pada lampiran 1. Adapun hasil *Displacement* maksimum ditampilkan pada tabel dan gambar di bawah ini:



Gambar 4.8 Perbandingan Grafik *Displacement* pada Node 2 Sumbu-z (a) Matlab dan (b) SAP 2000

Tabel 4.7 Perbandingan Hasil *Displacement* Maksimum pada Struktur Truss 2 Tingkat

Displ.	Proposed Software		Commercial Software		Δ	
	Max.	Min.	Max.	Min.	%	
2x	0.05491	-0.04916	0.05490	-0.04916	0.02133%	0.01479%
2y	0.00959	-0.00866	0.00958	-0.00866	0.02088%	0.01614%
3x	0.12506	-0.11258	0.12504	-0.11260	0.02057%	0.01497%
3y	0.00993	-0.00913	0.00993	-0.00913	0.01702%	0.01780%
4x	0.12043	-0.10837	0.12041	-0.10839	0.01999%	0.01540%
4y	0.02739	-0.02980	0.02739	-0.02979	0.01651%	0.01862%
5x	0.04567	-0.04096	0.04566	-0.04097	0.02069%	0.01392%
5y	0.01779	-0.01952	0.01779	-0.01952	0.01228%	0.01944%
6x	0.00000	0.00000	0.00000	0.00000	0.00997%	0.00920%

4.4. Penerapan Algoritma Simulasi *Monte Carlo* pada Algoritma Metode Elemen Hingga Model Dinamis beserta listing yang digunakan

4.4.1. Struktur *Plane Truss* Model Segitiga

Struktur *Plane Truss* model segitiga pada gambar 4.3 akan dihitung dengan data *mechanical properties* sebagai berikut:

Tabel 4.8 *Mechanical Properties* pada Struktur *Plane Truss* Model Segitiga

Variabel Input	Rata-rata	COV	Jenis Distribusi
<i>Modulus Young (E)</i>	29000000 lb/in ²	0.075	Normal
<i>Yield strength (σ_y)</i>	39600 psi	0.078	Normal
<i>Dispalacement Control (u_c)</i>	0.161281241 in	0.078	Normal
<i>Massa Tambah (M_{add})</i>	100000 lb	0.1	Normal
<i>Diameter (D)</i>	12.3 in	-	-
<i>Panjang Batang Horizontal (L1)</i>	118.11 in	-	-
<i>Panjang Batang Diagonal (L2)</i>	167.0328 in	-	-
<i>Massa jenis (ρ)</i>	0.7 lb/in ³	-	-

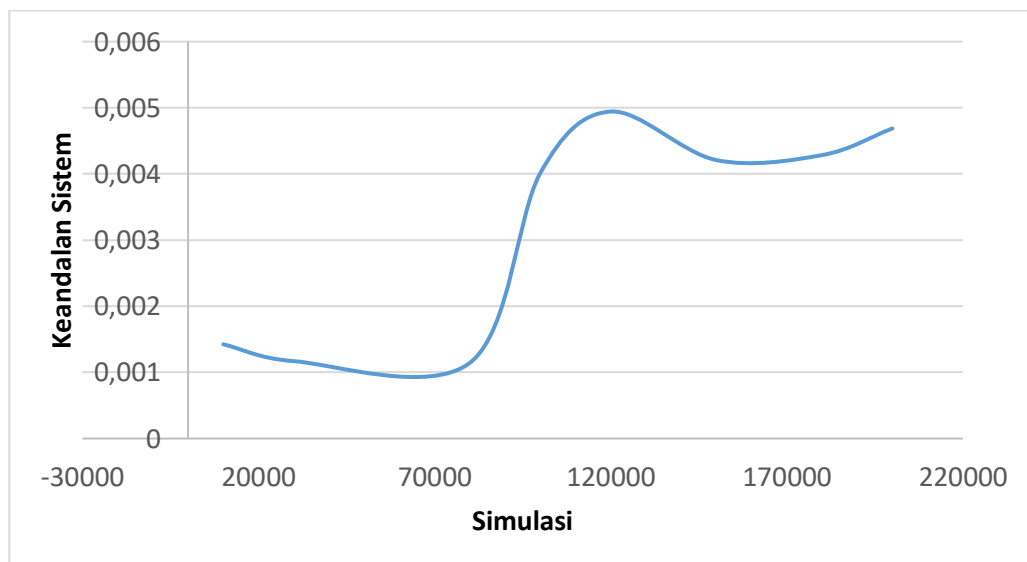
Adapun pembebanan yang digunakan tetap sama. Perhitungan ini, akan disimulasikan bertahap, sehingga akan diketahui *displacement* dan keandalan sistem pada masing-masing tahap simulasi. Hasil simulasi dapat dilihat pada Tabel 4.9 berikut ini:

Tabel 4.9 Hasil *Running* Struktur *Plane Truss* Model Segitiga

Jumlah Simulasi	Displacement Rata-rata pada Joint 2 (in)		Keandalan Sistem	$\frac{(u_{yi} + 1) - u_{yi}}{u_{yi}} \times 100\%$	$\frac{(R_i + 1) - R_i}{R_i} \times 100\%$	CPU Time
	ux	uz				
10000	0.015833	0.004963	0.702000	0.000632	0.001	143.607
30000	0.015817	0.004958	0.854533	0.000632	0.001	426.201
80000	0.015822	0.004959	0.874613	0.000632	0.001	1091.061
100000	0.015818	0.004958	0.249420	0.000632	0.004	1452.644
120000	0.015815	0.004957	0.202317	0.000632	0.005	1748.02
150000	0.015829	0.004961	0.237560	0.000632	0.004	2181.845
180000	0.015826	0.004960	0.233411	0.000632	0.004	2543.478
200000	0.015825	0.004960	0.213390	0.000632	0.005	2809.365

Dari tabel tersebut, dapat diplot hubungan jumlah simulasi dengan displacement dan keandalan yang terjadi. Sebagaimana yang ditampilkan pada gambar 4.9 di bawah ini:

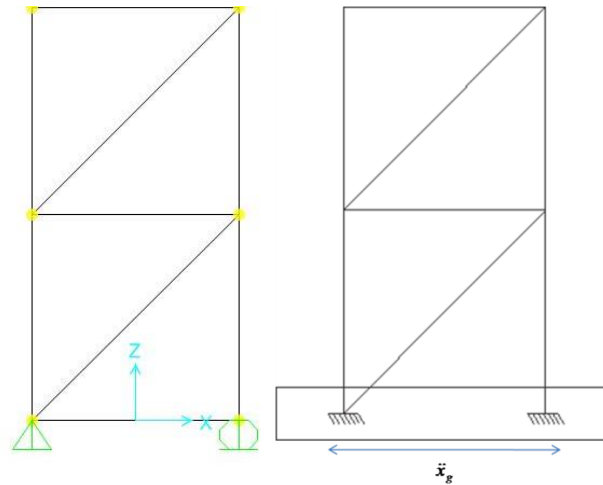
Gambar 4.9 Grafik Hubungan Jumlah Simulasi dengan Keandalan



Dari gambar 4.9 dan tabel 4.9 di atas menunjukkan bahwa simulasi yang dilakukan menghasilkan keandalan yang memenuhi syarat konvergen.

4.4.2. Struktur *Truss* 2 Tingkat

Model struktur kedua yang dibentuk adalah *Truss Element* 2 Tingkat yang ditunjukkan pada gambar di bawah ini:



Gambar 4.10 Model Elemen Truss 2D Hasil simulasi

Struktur di atas, mempunyai mechanical properties dan data pembebanan seperti yang ditunjukkan pada Tabel 4.10 sebagai berikut:

Tabel 4.10 *Material Properties* Struktur Truss 2 Tingkat

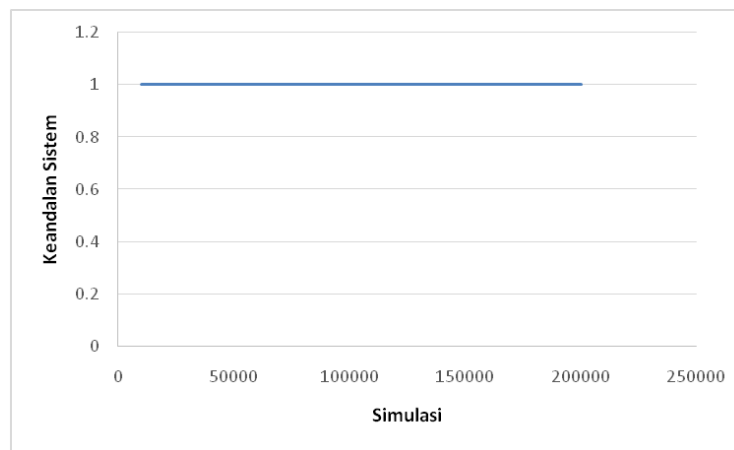
Variabel Input	Rata-rata	COV	Jenis Distribusi
<i>Modulus Young (E)</i>	29000000 lb/in ²	0.075	Normal
<i>Yield strength (σ_y)</i>	39600 psi	0.078	Normal
<i>Dispalacement Control (u_c)</i>	0.161281241 in	0.078	Normal
<i>Massa Tambah (M_{add})</i>	100000 lb	0.1	Normal
<i>Diameter (D)</i>	28.2 in	-	-
<i>Panjang Batang Horizontal (L1)</i>	118.11 in	-	-
<i>Panjang Batang Diagonal (L2)</i>	167.0328 in	-	-
<i>Massa jenis (ρ)</i>	0.7 lb/in ³	-	-

Dengan data-data di atas, struktur ini akan di *running* secara bertahap pula untuk mengetahui hasil analisa keandalan dan hubungan konvergensinya dengan jumlah simulasi yang dilakukan. Tabel 4.11 di bawah ini adalah hasil running dengan jumlah simulasi bertahap yang ditinjau hanya pada joint yang memiliki nilai kritis yaitu pada nodal 3:

Tabel 4.11 Menampilkan hasil *Displacement* terbesar pada nodal 3

Jumlah Simulasi	<i>Displacement Rata-rata pada Joint 3 (in)</i>		Keandalan Sistem	$\frac{(u_{yi} + 1) - u_{yi}}{u_{yi}} \times 100\%$	$\frac{(R_i + 1) - R_i}{R_i} \times 100\%$	CPU Time (sec)
	ux	Uz				
10000	0.018045786	0.001428306	1	0.000554	1	237.79
30000	0.018047677	0.001428465	1	0.000554	1	851.877
80000	0.017825059	0.001410791	1	0.000561	1	1658.075
100000	0.017823538	0.00141067	1	0.000561	1	2070.713
120000	0.017820781	0.001410446	1	0.000561	1	2465.141
150000	0.017829864	0.001411171	1	0.000561	1	3094.173
180000	0.018041233	0.001427954	1	0.000554	1	3734.573
200000	0.017827271	0.001410964	1	0.000561	1	4142.88

Dari tabel 4.11 tersebut juga diplot hubungan jumlah simulasi dengan displacement dan keandalan yang terjadi. Sebagaimana yang ditampilkan pada gambar 4.11 di bawah ini:



Gambar 4.11 Grafik Hubungan Jumlah Simulasi dengan Keandalan

Dari gambar 4.11 dan tabel 4.11 di atas menunjukkan bahwa simulasi yang dilakukan menghasilkan keandalan yang memenuhi syarat konvergen.

(Halaman ini sengaja dikosongkan)

BAB V
PENUTUP

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pemaparan mengenai aplikasi Simulasi Monte Carlo pada Metode Elemen Hingga untuk model dinamis struktur sederhana 2 dimensi di atas, maka dapat disimpulkan sebagai berikut:

1. *Software* yang telah dibuat dan telah dibandingkan dengan *SAP 2000 v.14* dengan menggunakan metode *time history* dengan teknik numerik *Newmark Method*, diperoleh error terbesar memiliki nilai 8.532% yang menunjukkan nilai yang relatif cukup kecil baik itu hasil dari frekuensi natural dan displacement yang terjadi pada model *Plane Truss* Segitiga dan 0.019% pada model *Plane Truss* 2 Tingkat.
2. Penerapan Simulasi *Monte Carlo* pada Metode Elemen Hingga solusi persamaan model dinamis telah dilakukan menggunakan algoritma yang telah dibuat. Algoritma ini dituangkan dalam bentuk listing fungsi dalam bahasa pemrograman *Matlab* versi 2011 yang mampu menganalisa keandalan struktur *Plane Truss* Model Dinamis.

5.2. Saran

Saran yang bisa penulis berikan dari tugas akhir ini adalah:

1. Teknik numerik yang digunakan dapat divariasikan seperti *central difference method*, *wilson theta method*, *interpolasi*, dan sebagainya dalam penyelesaian persamaan model dinamis struktur.
2. *Software* ini perlu dibandingkan menggunakan software berbayar lainnya seperti *SACS*, *ANSYS*, *GT-STRUDL*, *MSC-NASTRAN*, dll.
3. Penampilan dari *software* ini perlu diperbaiki dengan mode *GUI* agar lebih mudah digunakan.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- Campbell, S. L. 2010. **Modeling and Simulation in Matlab/Scicos with ScicosLab 4.4**. 2nd ed. London: Springer.
- Chapra, S. C. 2009. **Numerical Method for Engineer**. 6th ed. United States of America: McGraw-Hill.
- Cherng, Rwey-Hua. 1992. **Stochastic Finite Element Analyses of Uncertain Nonlinear Plane Under Random Excitatitions**. Urbana: Illinois. National Science Foundation.
- Chopra, Anil K. 2011. **Dynamics of Structures: Theory and Application to Earthquake Engineering**. Boston: USA. Pearson.
- Clough, Ray W & Joseph Penzen. 2003. **Dynamics of Structures**. Berkeley: USA. Computers & Structures, Inc.
- Craig, Roy. 2006. **Fundamental of Structural Dynamics**. New Jersey: Canada. John Wiley.
- Cook, Robert D. 1996. **Konsep dan Aplikasi Metode Elemen Hingga**. USA. John Wiley.
- Datta, T. K, 2010. **Seismic Analysis of Structures**. Singapore. John Wiley.
- DNV RP G101. 2002. **Risk Based Inspection of Offshore Topsides Static Mechanical Equipment**. Norway: DET NORSKE VERITAS.
- Dymiotis, C. 2001. **Reliability Based Code Calibration for Earthquake-Resistent Design**. Edinburgh. JCSS Workshop on Reliability Based Code Calibration.
- GB 50011. 2010. **Code for Seismic Design of Buildings**. China: National Standard of The Peoples's Republic of China.
- Guoliang, J., Chen Lin & Dong Jiamei. 1993. *Monte Carlo Finite Element Method of Structure Reliability Analysis*. **Reliability Engineering and Sistem Safety**. **40: 77 -83**.
- Haldar, A. & S. Mahadevan. 1991. Practical Random Field Discretization in Sthochastic Finite Element Analysis. **Structural Safety**, **9: 283-304**. Elsevier.
- Koduru, S.D. & Haukaas, T. 2010. Feasibility of FORM in finite element reliability analysis. **Structural Safety**, **32: 145-153**. Elsevier.
- Li, Jie & Jiabing Chen. 2009. **Stochastic Dynamics of Structures**. Singapore. John Wiley & Sons.

- Lutfi, MH. 2015. **Monte Carlo Finite Elemen Method berbasis *Matlab* pada analisa keandalan struktur *jacket* pada APN-A Platform**. Surabaya: Tugas Akhir Jurusan Teknik Kelautan FTK ITS.
- Logan, D. L. 1992. **A First Course in the Finite Element Method**. 2nd ed. Boston: PWS Publishing Company.
- Paz, Mario. 1990. **Dinamika Struktur Teori dan Perhitungan**. Edisi kedua. Jakarta: Indonesia. Erlangga.
- Ranganathan, R. 1990. **Reliability Analysis and Design of Structures**. Delhi: Tata McGraw Hill.
- Raychaudhuri, Samik. 2008. **Introduction to Monte Carlo Simulation**. Proceedings of 2008 Winter Simulation Conference. Broomfield: USA.
- Rosyid, D. M. 2007. **Pengantar Rekayasa Keandalan**. Surabaya: Airlangga University Press.
- Setiawan, A. 2015. **Analisa struktur**. Jakarta: Indonesia. Penerbit Erlangga.
- Schueller, G.I., Shinozuka, M. 1987. **Stochastic Methods in Structural Dynamics**. Dordrecht: Netherland. Marthinus Nijhoff Publisher.
- Wisudawan, A., Setyo, Y., Rosyid, D. 2013. *Development of Matlab-Based Structural Reliability Analysis Software Using Monte Carlo Simulated Finite Element Method*. **International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E 2013)**. Bandung: Indonesia.

LAMPIRAN 1

PERBANDINGAN DENGAN SOFTWARE

SAP 2000

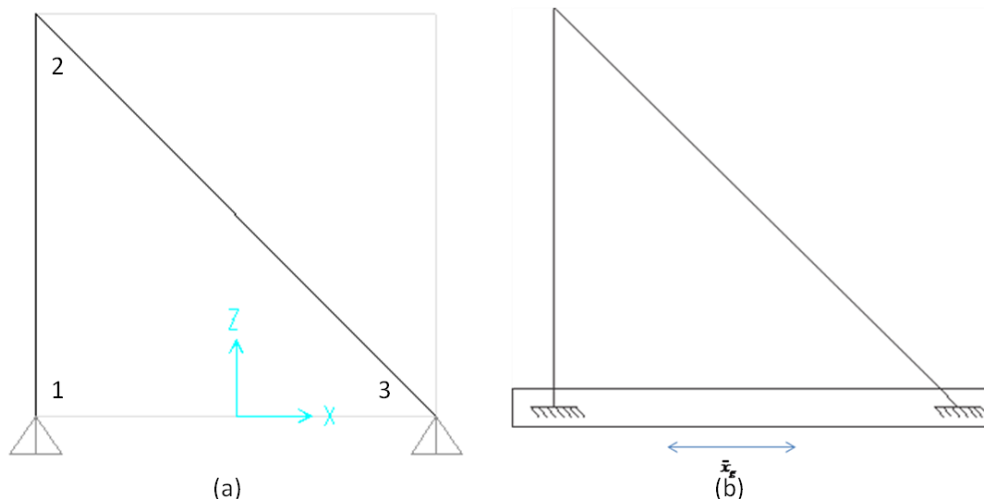
LAMPIRAN 1

PERBANDINGAN DENGAN SOFTWARE SAP 2000

I. Struktur *Plane Truss Element* Segitiga

Pada penelitian ini digunakan aplikasi Matlab versi 2011 dalam menentukan matriks modal, solusi persamaan dinamis untuk menentukan perpindahan struktur akibat beban eksternal, dan analisa keandalan. Proses perbandingan ini terbatas pada analisa matriks modal dan perpindahan mengingat hasil yang diberikan sama-sama analisa deterministik.

Variabel	Harga
Modulus Young (E)	29000000 lb/in ²
Luas Penampang (A)	12.3 in ²
Panjang Batang Horizontal (L1)	118.11 in
Panjang Batang Diagonal (L2)	167.0328 in
Massa jenis (ρ)	0.7 lb/in ³
Massa Tambah (M_{add})	100000 lb



Adapun solusi penyelesaian pada struktur *Plane Truss Element* di atas adalah dengan mengikuti *flowchart* pada pembahasan pada bab 4 akan dijelaskan sebagai berikut:

Langkah 1

Masukkan data material berupa *modulus young* (e), luas penampang (a), massa jenis (ρ) panjang batang vertikal (L1) dan panjang batang diagonal (L2) yang memiliki nilai sebagai berikut:

```
% Baja A36
e = 29*10^6 ;
rho = 7.345E-04 ;
d = 13.2 ;
```

```

r = d/2 ;
csa = r^2*pi() ;
l1 = 118.11 ;
l2 = 167.0328 ;

```

Karena batang yang ditinjau hanya ada 2, maka pemanggilan modul *stiffness* dilakukan sebanyak batang tersebut untuk dimasukkan data material dan sudut kemiringannya.

```

k1 = PlaneTrussElementStiffness(e, csa, l1, 90) ;
k2 = PlaneTrussElementStiffness(e, csa, l2, 315) ;

```

Maka saat dilakukan *running* pada simulasi akan menghasilkan matriks global pada model yang akan dianalisa.

k1 =

1.0e+07 *

0	0	0	0
0	3.360075779968478	0	-3.360075779968478
0	0	0	0
0	-3.360075779968478	0	3.360075779968478

k2 =

1.0e+07 *

1.187965927566552	-1.187965927566552	-1.187965927566552	1.187965927566552
-1.187965927566552	1.187965927566552	1.187965927566552	-1.187965927566552
-1.187965927566552	1.187965927566552	1.187965927566552	-1.187965927566552
1.187965927566552	-1.187965927566552	-1.187965927566552	1.187965927566552

Begitupun pada matriks massa maka pemanggilan modul massa dilakukan sebanyak batang tersebut.

```

m1 = PlaneTrussElementMass(rho, a, l1, 90) ;
m2 = PlaneTrussElementMass(rho, a, l2, 315) ;

```

Maka saat dilakukan *running* pada simulasi akan menghasilkan matriks lokal pada model yang akan dianalisa.

m1 =

5.935895104461309	0	0	0
0	5.935895104461309	0	0
0	0	5.935895104461309	0

0	0	0	5.935895104461309
---	---	---	-------------------

m2 =

8.394625178261492	0	0	0
0	8.394625178261492	0	0
0	0	8.394625178261492	0
0	0	0	8.394625178261492

Pada tahap ini juga dibentuk matriks gabungan. Untuk memudahkan proses penggabungan dilakukan pemanggilan matriks nol sebesar ukuran matriks awal sebagai pijakan ukuran matriks penggabungan. Hal ini ditentukan dengan melihat banyaknya *degree of freedom* pada struktur yang sedang dianalisa. Listing fungsi yang digunakan adalah sebagai berikut:

```
%Properti Finite Element
K = zeros(6,6) ;
M = zeros(6,6) ;
```

Output dari listing fungsi di atas adalah sebagai berikut:

```
K =

    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0

M =

    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
```

Selanjutnya melakukan pemanggilan listing fungsi yang menggabungkan matriks, yaitu sebagai berikut:

```
% Assemble matrix element
K = PlaneTrussAssemble(K,k1,1,2) ;
K = PlaneTrussAssemble(K,k2,2,3) ;
M = PlaneTrussAssemble(M,m1,1,2) ;
M = PlaneTrussAssemble(M,m2,2,3) ;
```

Maka *output* dari listing fungsi di atas adalah sebagai berikut:

K =

1.0e+07 *

0	0	0	0	0	0
0	3.3601	0	-3.3601	0	0
0	0	0	0	0	0
0	-3.3601	0	3.3601	0	0
0	0	0	0	0	0
0	0	0	0	0	0

K =

1.0e+07 *

0	0	0	0	0	0
0	3.3601	0	-3.3601	0	0
0	0	1.1880	-1.1880	-1.1880	1.1880
0	-3.3601	-1.1880	4.5480	1.1880	-1.1880
0	0	-1.1880	1.1880	1.1880	-1.1880
0	0	1.1880	-1.1880	-1.1880	1.1880

Penggabungan matriks ini akan menyesuaikan pengisian kolom dan baris berdasarkan urutan *degree of freedom* yang direpresentasikan oleh model, sehingga didapatkan:

K =

1.0e+07 *

0	0	0	0	0	0
0	3.3601	0	-3.3601	0	0
0	0	1.1880	-1.1880	-1.1880	1.1880
0	-3.3601	-1.1880	4.5480	1.1880	-1.1880
0	0	-1.1880	1.1880	1.1880	-1.1880
0	0	1.1880	-1.1880	-1.1880	1.1880

Untuk matriks massa global dari setiap batang ditampilkan sebagai berikut:

M =

5.9359	0	0	0	0	0
--------	---	---	---	---	---

0	5.9359	0	0	0	0
0	0	5.9359	0	0	0
0	0	0	5.9359	0	0
0	0	0	0	0	0
0	0	0	0	0	0

M =

5.9359	0	0	0	0	0
0	5.9359	0	0	0	0
0	0	14.3305	0	0	0
0	0	0	14.3305	0	0
0	0	0	0	8.3946	0
0	0	0	0	0	8.3946

Setelah melalui penggabungan matriks massa didapatkan:

M =

5.9359	0	0	0	0	0
0	5.9359	0	0	0	0
0	0	14.3305	0	0	0
0	0	0	14.3305	0	0
0	0	0	0	8.3946	0
0	0	0	0	0	8.3946

Langkah 2

Melakukan analisa modal menggunakan *listing* yang dijelaskan sebelumnya dengan melalui beberapa tahapan yang akan dijelaskan sebagai berikut:

- Masukkan matriks kekakuan dan matriks massa struktur yang telah ditentukan kondisi batasnya dari *listing* fungsi matriks penggabungan. Karena terdapat nodal yang memiliki tumpuan sendi (menyebabkan gerak translasi x dan y bernilai nol), maka berlaku pada nodal yang terdapat sendi adalah:

$$d_{1x}=d_{1y}=d_{3x}=d_{3y}=0$$

Sehingga matriks penggabungan tereduksi menjadi ukuran matriks 2×2 berdasarkan joint yang tidak dikekang oleh tumpuan apapun. Pemanggilan listing yang dilakukan adalah sebagai berikut:

```
% Apply boundary conditions for stiffness matrix and mass
K = K(3:4,3:4) ;
M = M(3:4,3:4) ;
```

Maka, output dari matriks diatas adalah sebagai berikut:

K =

```
81.0270 -0.0000
0 492.4916
```

M =

```
(1.0e+05)*
1.0001 0
0 1.0001
```

- b. Tentukan nilai moda getar dan frekuensi natural dengan menggunakan *toolbox* “eig” pada matlab, dimana *toolbox* ini berguna untuk mengeluarkan nilai eigen atau frekuensi natural dan vektor eigen atau moda getar dari matriks kekakuan dan massa sebelum ini, adapun listing yang dipanggil sebagai berikut:

```
% Modal Analysis
[V,D] = eig(K,M)
```

Setelah melakukan *running*, maka hasil yang didapatkan digambarkan sebagai berikut:

V =

```
-0.0030 -0.0010
-0.0010 0.0030
```

D =

```
81.0270 0
0 492.4916
```

Langkah 3

Normalisasi moda getar V dilakukan untuk mendapatkan frekuensi natural yang dikuadratkan atau disimbolkan dengan omega (Ω). Sehingga untuk memperoleh frekuensi natural dilakukan akar kuadrat Ω terlebih dahulu. Adapun listing fungsi yang dilakukan adalah sebagai berikut:

```
Factor=diag(V'*M*V);

Phi=V*inv(sqrt(diag(Factor))) %normalisasi mode shape
Omega=diag(sqrt(Phi'*K*Phi)) %cek orthogonal
```

Output dari listing fungsi di atas adalah sebagai berikut:

Phi =

```
-0.0030 -0.0010
-0.0010  0.0030
```

Omega =

```
9.0015
22.1922
```

Langkah 4

Mode shape perlu dilakukan cek orthogonalitas dengan mengalikan modal matriks yang ternormalisasi dengan matriks M untuk menghasilkan matriks identitas sebagai bentuk massa tergeneralisasi.

```
MM = phi'*M*phi
```

Setelah itu dapat dilihat hasil nilai massa sebagai berikut:

```
MM =
    1.0000    0
         0    1.0000
```

Matriks massa di atas menunjukkan matriks yang orthogonal karena memiliki bentuk matriks identitas.

Langkah 5

Setelah itu didapatkan nilai modal matriks atau moda getar pada struktur yang memenuhi syarat orthogonal. Jika tidak akan dilakukan pengulangan pada tahap evaluasi persamaan *undamped free vibration*. Pada simulasi ini modal matriks atau moda getar disimbolkan dengan notasi Phi:

Phi =

```
-0.0030 -0.0010
-0.0010  0.0030
```

Langkah 6

Nilai matriks modal yang didapatkan akan digunakan pada tahapan pembentukan matriks massa, kekakuan, redaman dan eksitasi dengan bentuk tergeneralisasi.

Untuk redaman sendiri diasumsikan sebagai redaman klasik jenis *rayleigh damping*. Untuk memanggil output persamaanya dilakukan dengan listing sebagai berikut ini:

```
% Damping
aa = [1/Omega(1) Omega(1);1/Omega(2) Omega(2)] ;
bb = inv(aa)*2*[0.05;0.05] ;
z = bb(1)*M+bb(2)*K ;
C = phi'*z*phi ;
```

Sedangkan *output* dari listing diatas adalah sebagai berikut:

```
C =

    0.6404    -0.0000
   -0.0000    0.6404
```

Sedangkan untuk percepatan ditentukan dengan memasukkan *influence vector*, H, dan gaya setelah dikalikan dengan modal matriks yang terjadi pada struktur:

```
% Excitation
H = [1;0] ;
p = -M*H*g8' ;
P = phi'*p ;
r = [g8'] ;
```

Langkah 7

Merupakan tahap perhitungan awal pada analisa dinamis struktur dengan metode *Newmark*, tahapan ini terbagi lagi beberapa langkah yang dijelaskan sebagai berikut:

- i. Masukkan kondisi awal pada kondisi sistem tidak mengalami perpindahan, perpindahan, perpindahan modal, dan kecepatan modal awal bernilai nol, sehingga dituliskan pada matlab adalah sebagai berikut:

```
depl = zeros(n,nt) ;
vel = zeros(n,nt) ;
accl = zeros(n,nt) ;
U = zeros(n,nt) ;

% Initial Conditions
depl(:,1) = zeros ;
vel(:,1) = zeros ;
U(:,1) = phi*depl(:,1) ;
```


Yang menghasilkan *output* berupa nilai 0 berukuran dengan baris yang sejumlah banyaknya data dan kolom sejumlah ukuran matriks massa yang terbentuk.

- ii. Masukkan persamaan gaya awal yang bernilai nol.

```
P0 = zeros(n,1) ;
```

- iii. Masukkan percepatan modal awal yang bernilai nol

```
accl(:,1) = MM\ (P0-C*vel(:,1)-K*depl(:,1))
```

- iv. Tentukan *time step* yang digunakan

Pada penelitian ini dipilih *time step*, $\Delta t = 0.03$

- v. Masukkan matriks M, C, $\Delta t = 0.03$, $\gamma = \frac{1}{2}$ dan $\beta = \frac{1}{4}$ dengan fungsi listing:

```
% Constants used in Newmark's integration
a1 = gaama/(beta*dt) ; a2 = 1/(beta*dt^2) ; %a1
a3 = 1/(beta*dt) ; a4 = (gaama/beta)-1 ; %a2
a5 = (1/(2*beta))-1 ; a6 = dt*((gaama/(2*beta))-1) ; %a3

a = a1*C+a2*MM %a1
b = a3*MM+a4*C %a2
c = a5*MM+a6*C %a3
```

sehingga didapatkan:

a =

```
1.0e+03 *
1.1325 -0.0000
-0.0000 1.1325
```

b =

```
67.3071 -0.0000
-0.0000 67.3071
```

c =

```
1.0000 0
0 1.0000
```

- vi. Masukkan nilai K dan a_1 dengan menggunakan fungsi *listing*:

```
Kcap = K+a %1.6
```

sehingga didapatkan

```
Kcap =

1.0e+006 *

    0.5769    -0.0034
   -0.0034     3.4581
```

Langkah 8

Perhitungan untuk setiap langkah waktu i sampai batas sejumlah data yang telah ditentukan (sebanyak 2000 data dengan $\Delta t = 0.03$)

- Selesaikan persamaan untuk menentukan eksitasi pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
% Time step starts
for i = 1:1
    Pk = P(1:lambda,i+1) %2.1
    delP = Pk+a*depl(:,i)+b*vel(:,i)+c*accl(:,i) %2.1
```

Sehingga menghasilkan nilai eksitasi pada waktu $i=1$ adalah sebagai berikut:

```
delP =

176.4724
27.1636
```

- Selesaikan perpindahan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
depl(:,i+1) = Kcap\delP %2.2
```

Sehingga menghasilkan keluaran perpindahan modal pada waktu $i=1$ adalah sebagai berikut:

```
depl =

0 0.0027
0 0.0006
```

- Selesaikan kecepatan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
vel(:,i+1) = a1*(depl(:,i+1)-depl(:,i))-a4*vel(:,i)-a6*accl(:,i)
```

Sehingga menghasilkan kecepatan modal pada waktu $i=1$ adalah sebagai berikut:

```
vel(:,2)

0.0893
0.0212
```

- d. Selesaikan percepatan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
accl(:,i+1) = a2*(depl(:,i+1)-depl(:,i))-a3*vel(:,i)-a5*accl(:,i)
```

Sehingga menghasilkan percepatan modal pada waktu $i=1$ adalah sebagai berikut:

accl(:,2)

2.9770

0.7066

- e. Selesaikan perpindahan total pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
U(:,i+1) = phi*depl(:,i+1) ; %2.5
end
```

Sehingga menghasilkan perpindahan total pada waktu $i=1$ adalah sebagai berikut:

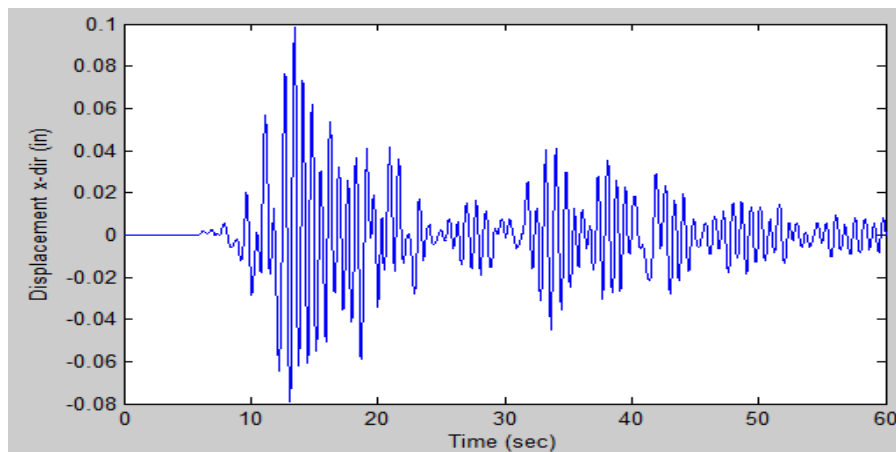
U(:,2)

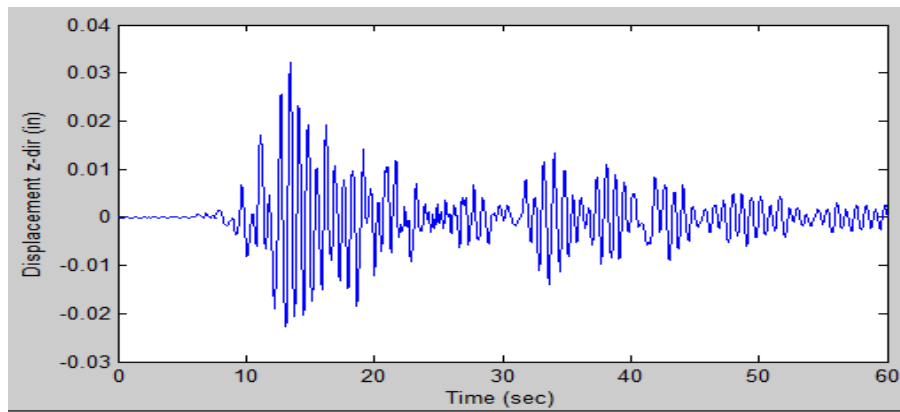
1.0e-05 *

-0.8683

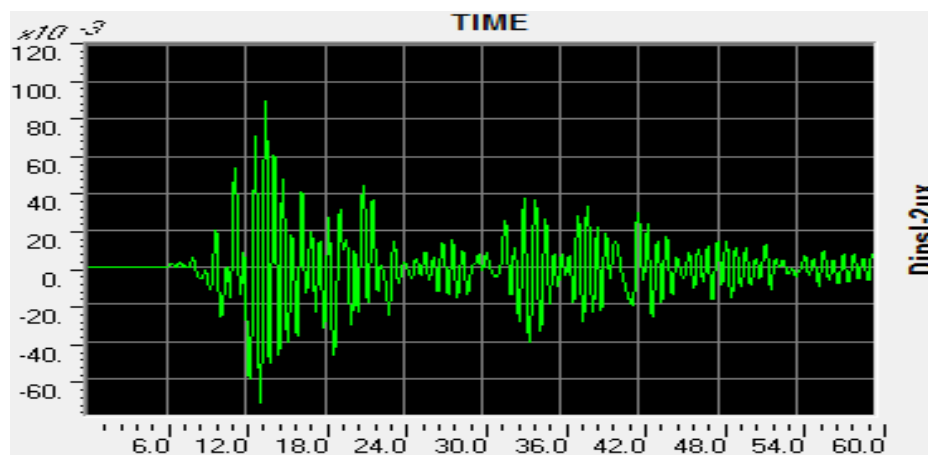
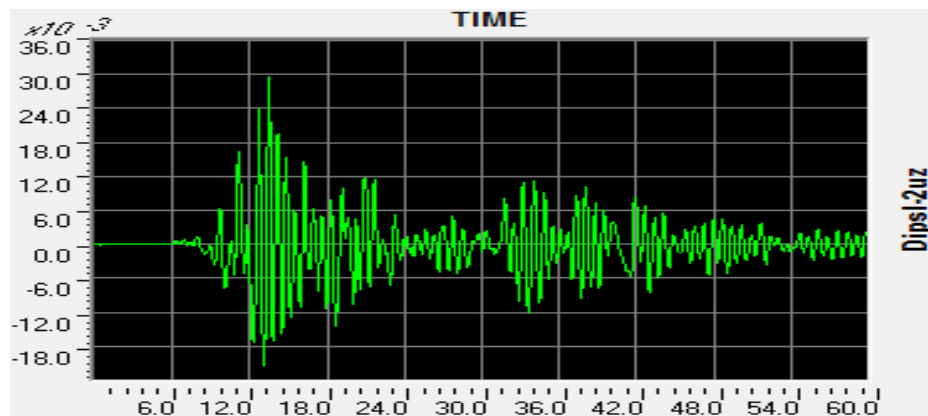
-0.0650

- f. Hasil dari perpindahan total dapat ditampilkan pada tabel dan di plotkan pada gambar berdasarkan fungsi waktu.
- Hasil Grafik Displacement pada Matlab sepanjang waktu respon percepatan gempa





- Hasil Grafik Displacement pada SAP 2000 sepanjang waktu respon percepatan gempa

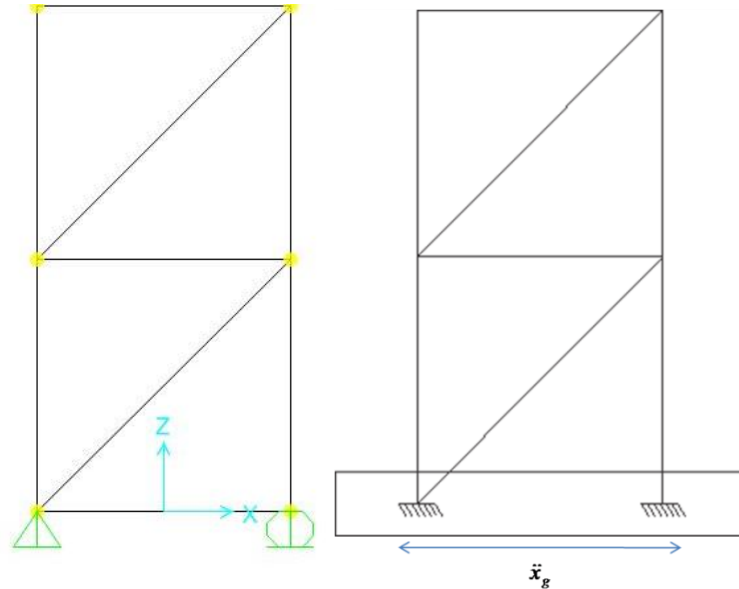


II. Struktur *Plane Truss* 2 Tingkat

Pada model ke dua ini tidak begitu berbeda jauh dengan model pertama, hanya saja terdapat beberapa *member* yang ditambahkan pada struktur.

Variabel	Harga
Modulus Young (E)	29000000 lb/in ²
Diameter (D)	

Panjang Batang Horizontal (L1)	118.11 in
Panjang Batang Diagonal (L2)	167.0328 in
Massa jenis (ρ)	0.7 lb/in ³
Massa Tambah (M_{add})	100000 lb



Solusi penyelesaian pada struktur *Plane Truss Element* akan dijelaskan sebagai berikut:

Langkah 1

```
% Baja A36
e = 29*10^6 ;
rho = 7.345E-04 ;
d = 28.2 ;
r = d/2 ;
csa = r^2*pi() ;
l1 = 118.11 ;
l2 = 167.0328 ;
```

Karena batang yang ditinjau hanya ada 9, maka pemanggilan modul matriks kekakuan dan matriks massa dilakukan sebanyak batang tersebut untuk dimasukkan data material dan sudut kemiringannya.

```
k1 = PlaneTrussElementStiffness(e,csa,l1,0) ; % k1,1,6
m1 = PlaneTrussElementMass(rho,csa,l1,0) ;
k2 = PlaneTrussElementStiffness(e,csa,l1,90) ; % k2,1,2
m2 = PlaneTrussElementMass(rho,csa,l1,90) ;
k3 = PlaneTrussElementStiffness(e,csa,l2,45) ; % k3,1,5
m3 = PlaneTrussElementMass(rho,csa,l2,45) ;
k4 = PlaneTrussElementStiffness(e,csa,l1,270) ; % k4,5,6
m4 = PlaneTrussElementMass(rho,csa,l1,270) ;
k5 = PlaneTrussElementStiffness(e,csa,l1,0) ; % k5,2,5
m5 = PlaneTrussElementMass(rho,csa,l1,0) ;
k6 = PlaneTrussElementStiffness(e,csa,l1,90) ; % k6,2,3
m6 = PlaneTrussElementMass(rho,csa,l1,90) ;
k7 = PlaneTrussElementStiffness(e,csa,l2,45) ; % k7,2,4
m7 = PlaneTrussElementMass(rho,csa,l2,45) ;
```

```
k8 = PlaneTrussElementStiffness(e, csa, 11, 270) ; % k8, 4, 5
m8 = PlaneTrussElementMass(rho, csa, 11, 270) ;
k9 = PlaneTrussElementStiffness(e, csa, 11, 0) ; % k9, 3, 4
m9 = PlaneTrussElementMass(rho, csa, 11, 0) ;
```

Maka saat dilakukan *running* pada simulasi akan menghasilkan matriks global pada model yang akan dianalisa.

Selanjutnya melakukan pemanggilan listing fungsi yang menggabungkan matriks, yaitu sebagai berikut:

```
K = PlaneTrussAssemble(K, k1, 1, 6) ;
K = PlaneTrussAssemble(K, k2, 1, 2) ;
K = PlaneTrussAssemble(K, k3, 1, 5) ;
K = PlaneTrussAssemble(K, k4, 5, 6) ;
K = PlaneTrussAssemble(K, k5, 2, 5) ;
K = PlaneTrussAssemble(K, k6, 2, 3) ;
K = PlaneTrussAssemble(K, k7, 2, 4) ;
K = PlaneTrussAssemble(K, k8, 4, 5) ;
K = PlaneTrussAssemble(K, k9, 3, 4) ;

M = PlaneTrussAssemble(M, m1, 1, 6) ;
M = PlaneTrussAssemble(M, m2, 1, 2) ;
M = PlaneTrussAssemble(M, m3, 1, 5) ;
M = PlaneTrussAssemble(M, m4, 5, 6) ;
M = PlaneTrussAssemble(M, m5, 2, 5) ;
M = PlaneTrussAssemble(M, m2, 2, 3) ;
M = PlaneTrussAssemble(M, m3, 2, 4) ;
M = PlaneTrussAssemble(M, m4, 4, 5) ;
M = PlaneTrussAssemble(M, m5, 3, 4) ;
```

Penggabungan matriks ini akan menyesuaikan pengisian kolom dan baris berdasarkan urutan *degree of freedom* yang direpresentasikan oleh model, sehingga didapatkan:

K =

```
1.0e+08 *
2.0757  0.5422  0  0  0  0  0  0 -0.5422 -0.5422 -1.5336  0
0.5422  2.0757  0 -1.5336  0  0  0  0 -0.5422 -0.5422  0  0
0  0  2.0757  0.5422  0  0 -0.5422 -0.5422 -1.5336  0  0  0
0 -1.5336  0.5422  3.6093  0 -1.5336 -0.5422 -0.5422  0  0  0  0
0  0  0  0  1.5336  0 -1.5336  0  0  0  0  0
0  0  0 -1.5336  0  1.5336  0  0  0  0  0  0
0  0 -0.5422 -0.5422 -1.5336  0  2.0757  0.5422  0  0  0  0
0  0 -0.5422 -0.5422  0  0  0.5422  2.0757  0 -1.5336  0  0
```

-0.5422	-0.5422	-1.5336	0	0	0	0	0	2.0757	0.5422	0	0
-0.5422	-0.5422	0	0	0	0	0	-1.5336	0.5422	3.6093	0	-1.5336
-1.5336	0	0	0	0	0	0	0	0	0	1.5336	0
0	0	0	0	0	0	0	0	0	-1.5336	0	1.5336

Dan matriks massa didapatkan:

M =

92.4969	0	0	0	0	0	0	0	0	0	0	0
0	92.4969	0	0	0	0	0	0	0	0	0	0
0	0	119.5886	0	0	0	0	0	0	0	0	0
0	0	0	119.5886	0	0	0	0	0	0	0	0
0	0	0	0	54.1834	0	0	0	0	0	0	0
0	0	0	0	0	54.1834	0	0	0	0	0	0
0	0	0	0	0	0	92.4969	0	0	0	0	0
0	0	0	0	0	0	0	92.4969	0	0	0	0
0	0	0	0	0	0	0	0	119.5886	0	0	0
0	0	0	0	0	0	0	0	0	119.5886	0	0
0	0	0	0	0	0	0	0	0	0	54.1834	0
0	0	0	0	0	0	0	0	0	0	0	54.1834

Langkah 2

Melakukan analisa modal menggunakan *listing* yang dijelaskan sebelumnya dengan melalui beberapa tahapan yang akan dijelaskan sebagai berikut:

- Masukkan matriks kekakuan dan matriks massa struktur yang telah ditentukan kondisi batasnya dari *listing* fungsi matriks penggabungan. Karena terdapat nodal yang memiliki tumpuan sendi (menyebabkan gerak translasi x dan y bernilai nol), maka berlaku pada nodal yang terdapat sendi adalah:

$$d_{1x}=d_{1y}=d_{3x}=d_{3y}=0$$

Sehingga matriks penggabungan tereduksi menjadi ukuran matriks 2×2 berdasarkan joint yang tidak dikekang oleh tumpuan apapun. Pemanggilan *listing* yang dilakukan adalah sebagai berikut:

```
K = K(3:11,3:11) ;
M = M(3:11,3:11) + diag([0;0;Madd;Madd;Madd;Madd;0;0;0]) ;
```

Maka, output dari matriks diatas adalah sebagai berikut:

K =

1.0e+08 *

2.0757	0.5422	0	0	-0.5422	-0.5422	-1.5336	0	0
0.5422	3.6093	0	-1.5336	-0.5422	-0.5422	0	0	0
0	0	1.5336	0	-1.5336	0	0	0	0
0	-1.5336	0	1.5336	0	0	0	0	0
-0.5422	-0.5422	-1.5336	0	2.0757	0.5422	0	0	0
-0.5422	-0.5422	0	0	0.5422	2.0757	0	-1.5336	0
-1.5336	0	0	0	0	0	2.0757	0.5422	0
0	0	0	0	0	-1.5336	0.5422	3.6093	0
0	0	0	0	0	0	0	0	1.5336

M =

1.0e+05 *

0.0012	0	0	0	0	0	0	0	0
0	0.0012	0	0	0	0	0	0	0
0	0	1.0005	0	0	0	0	0	0
0	0	0	1.0005	0	0	0	0	0
0	0	0	0	1.0009	0	0	0	0
0	0	0	0	0	1.0009	0	0	0
0	0	0	0	0	0	0.0012	0	0
0	0	0	0	0	0	0	0.0012	0
0	0	0	0	0	0	0	0	0.0005

- b. Tentukan nilai moda getar dan frekuensi natural dengan menggunakan *toolbox* “eig” pada matlab, dimana *toolbox* ini berguna untuk mengeluarkan nilai eigen atau frekuensi natural dan vektor eigen atau moda getar dari matriks kekakuan dan massa sebelum ini, adapun listing yang dipanggil sebagai berikut:

```
% Modal Analysis
[V,D] = eig(K,M)
```


Setelah melakukan *running*, maka hasil yang didapatkan digambarkan sebagai berikut:

V =

-0.0010	-0.0005	-0.0014	0.0015	-0.0637	-0.0457	0	-0.0109	-0.0457
-0.0002	0.0015	0.0002	0.0001	0.0109	0.0457	0	-0.0637	-0.0458
-0.0022	0.0000	-0.0007	-0.0021	-0.0000	-0.0000	0	-0.0000	-0.0000
-0.0002	0.0030	0.0010	-0.0001	-0.0000	-0.0000	0	0.0000	0.0000
-0.0022	0.0000	-0.0001	0.0023	0.0001	0.0000	0	0.0000	0.0000
0.0005	0.0010	-0.0029	0.0004	0.0000	0.0000	0	0.0000	-0.0000
-0.0008	-0.0005	-0.0007	0.0011	-0.0637	0.0457	0	-0.0109	0.0457
0.0004	0.0005	-0.0011	-0.0000	0.0109	-0.0457	0	-0.0638	0.0457
0	0	0	0	0	0	0.1359	0	0

D =

1.0e+06 *

0.0001	0	0	0	0	0	0	0	0
0	0.0008	0	0	0	0	0	0	0
0	0	0.0013	0	0	0	0	0	0
0	0	0	0.0032	0	0	0	0	0
0	0	0	0	0.3759	0	0	0	0
0	0	0	0	0	2.5651	0	0	0
0	0	0	0	0	0	2.8303	0	0
0	0	0	0	0	0	0	3.0969	0
0	0	0	0	0	0	0	0	3.4717

Langkah 3

Normalisasi moda getar V dilakukan untuk mendapatkan frekuensi natural yang dikuadratkan atau disimbolkan dengan omega (Ω). Sehingga untuk memperoleh frekuensi natural dilakukan akar kuadrat Ω terlebih dahulu. Adapun listing fungsi yang dilakukan adalah sebagai berikut:

```
Factor=diag(V'*M*V);
```

```
Phi=V*inv(sqrt(diag(Factor))) %normalisasi mode shape
```

```
Omega=diag(sqrt(Phi'*K*Phi)) %cek orthogonal
```

Output dari listing fungsi di atas adalah sebagai berikut:

Phi =

-0.0010	-0.0005	-0.0014	0.0015	-0.0637	-0.0457	0	-0.0109	-0.0457
-0.0002	0.0015	0.0002	0.0001	0.0109	0.0457	0	-0.0637	-0.0458
-0.0022	0.0000	-0.0007	-0.0021	-0.0000	-0.0000	0	-0.0000	-0.0000
-0.0002	0.0030	0.0010	-0.0001	-0.0000	-0.0000	0	0.0000	0.0000
-0.0022	0.0000	-0.0001	0.0023	0.0001	0.0000	0	0.0000	0.0000
0.0005	0.0010	-0.0029	0.0004	0.0000	0.0000	0	0.0000	-0.0000
-0.0008	-0.0005	-0.0007	0.0011	-0.0637	0.0457	0	-0.0109	0.0457
0.0004	0.0005	-0.0011	-0.0000	0.0109	-0.0457	0	-0.0638	0.0457
0	0	0	0	0	0	0.1359	0	0

Omega =

1.0e+03 *

0.0076

0.0277

0.0357

0.0565

0.6131

1.6016

1.6824

1.7598

1.8633

Langkah 4

Cek orthogonalitas.

$MM = \phi' * M * \phi$

Setelah itu dapat dilihat hasil nilai massa sebagai berikut:

MM =

1.0000	0.0000	0.0000	-0.0000	-0.0000	0.0000	0	-0.0000	-0.0000
0.0000	1.0000	0.0000	-0.0000	0.0000	-0.0000	0	-0.0000	-0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0	0.0000	0.0000

-0.0000	-0.0000	0.0000	1.0000	-0.0000	-0.0000	0	0.0000	0.0000
-0.0000	0.0000	0.0000	-0.0000	1.0000	-0.0000	0	0.0000	0.0000
0.0000	-0.0000	0.0000	-0.0000	-0.0000	1.0000	0	0.0000	0.0000
0	0	0	0	0	0	1.0000	0	0
-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000	0	1.0000	-0.0000
-0.0000	-0.0000	0.0000	0.0000	0.0000	0	0	-0.0000	1.0000

Matriks massa di atas menunjukkan matriks yang orthogonal karena memiliki bentuk matriks identitas.

Langkah 5

Setelah itu didapatkan nilai modal matriks atau moda getar pada struktur yang memenuhi syarat orthogonal. Jika tidak akan dilakukan pengulangan pada tahap evaluasi persamaan *undamped free vibration*. Pada simulasi ini modal matriks atau moda getar disimbolkan dengan notasi Phi:

Phi =

-0.0010	-0.0005	-0.0014	0.0015	-0.0637	-0.0457	0	-0.0109	-0.0457
-0.0002	0.0015	0.0002	0.0001	0.0109	0.0457	0	-0.0637	-0.0458
-0.0022	0.0000	-0.0007	-0.0021	-0.0000	-0.0000	0	-0.0000	-0.0000
-0.0002	0.0030	0.0010	-0.0001	-0.0000	-0.0000	0	0.0000	0.0000
-0.0022	0.0000	-0.0001	0.0023	0.0001	0.0000	0	0.0000	0.0000
0.0005	0.0010	-0.0029	0.0004	0.0000	0.0000	0	0.0000	-0.0000
-0.0008	-0.0005	-0.0007	0.0011	-0.0637	0.0457	0	-0.0109	0.0457
0.0004	0.0005	-0.0011	-0.0000	0.0109	-0.0457	0	-0.0638	0.0457
0	0	0	0	0	0	0.1359	0	0

Langkah 6

Nilai matriks modal yang didapatkan akan digunakan pada tahapan pembentukan matriks massa, kekakuan, redaman dan eksitasi dengan bentuk tergeneralisasi.

Untuk redaman sendiri diasumsikan sebagai redaman klasik jenis *rayleigh damping*. Untuk memanggil output persamaanya dilakukan dengan listing sebagai berikut ini:

% Damping

```

aa = [1/Omega(1) Omega(1);1/Omega(2) Omega(2)] ;
bb = inv(aa)*2*[0.05;0.05] ;
z = bb(1)*M+bb(2)*K ;
C = phi'*z*phi ;

```

Sedangkan *output* dari listing diatas adalah sebagai berikut:

C = 1.0e+03 *

```

0.0008  0.0000 -0.0000 -0.0000 -0.0000 -0.0000    0  0.0000 -0.0000
0.0000  0.0028 -0.0000  0.0000  0.0000 -0.0000    0 -0.0000 -0.0000
-0.0000 -0.0000  0.0042  0.0000 -0.0000  0.0000    0 -0.0000  0.0000
-0.0000  0.0000  0.0000  0.0097  0.0000  0.0000    0  0.0000  0.0000
-0.0000  0.0000 -0.0000  0.0000  1.0667 -0.0000    0  0.0000  0.0000
-0.0000 -0.0000  0.0000  0.0000 -0.0000  7.2750    0  0.0000  0.0000
      0      0      0      0      0      0  8.0271      0      0
0.0000 -0.0000 -0.0000  0.0000  0.0000  0.0000    0  8.7830      0
-0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000    0 -0.0000  9.8460

```

Sedangkan untuk percepatan ditentukan dengan memasukkan *influence vector*, H, dan gaya setelah dikalikan dengan modal matriks yang terjadi pada struktur:

```

% Excitation
H = [1;0;1;0;1;0;1;0;1];
p = -M*H*g8' ;
P = phi'*p ;
s = [g8'] ;

```

Langkah 7

Merupakan tahap perhitungan awal pada analisa dinamis struktur dengan metode *Newmark*, tahapan ini terbagi lagi beberapa langkah yang dijelaskan sebagai berikut:

- i. Masukkan kondisi awal pada kondisi sistem tidak mengalami perpindahan, perpindahan, perpindahan modal, dan kecepatan modal awal bernilai nol, sehingga dituliskan pada matlab adalah sebagai berikut:

```

depl = zeros(n,nt) ;
vel = zeros(n,nt) ;
accl = zeros(n,nt) ;
U = zeros(n,nt) ;

```

```
% Initial Conditions
depl(:,1) = zeros ;
vel(:,1) = zeros ;
U(:,1) = phi*depl(:,1) ;
```

Yang menghasilkan *output* berupa nilai 0 berukuran dengan baris yang sejumlah banyaknya data dan kolom sejumlah ukuran matriks massa yang terbentuk.

- ii. Masukkan persamaan gaya awal yang bernilai nol.

```
P0 = zeros(n,1) ;
```

- iii. Masukkan percepatan modal awal yang bernilai nol

```
accl(:,1) = MM\ (P0-C*vel(:,1)-K*depl(:,1))
```

- iv. Tentukan *time step* yang digunakan

Pada penelitian ini dipilih *time step*, $\Delta t = 0.03$

- v. Masukkan matriks M, C, $\Delta t = 0.03$, $\gamma = \frac{1}{2}$ dan $\beta = \frac{1}{4}$ dengan fungsi listing:

```
% Constants used in Newmark's integration
a1 = gaama/(beta*dt) ; a2 = 1/(beta*dt^2) ; %a1
a3 = 1/(beta*dt) ; a4 = (gaama/beta)-1 ; %a2
a5 = (1/(2*beta))-1 ; a6 = dt*((gaama/(2*beta))-1) ; %a3

a = a1*C+a2*MM %a1
b = a3*MM+a4*C %a2
c = a5*MM+a6*C %a3
```

sehingga didapatkan:

a =

```
1.0e+05 *
    0.0114    0.0000   -0.0000   -0.0000   -0.0000   -0.0000         0    0.0000   -0.0000
    0.0000    0.0120   -0.0000   -0.0000    0.0000   -0.0000         0   -0.0000   -0.0000
   -0.0000   -0.0000    0.0125    0.0000   -0.0000    0.0000         0   -0.0000    0.0000
   -0.0000   -0.0000    0.0000    0.0143    0.0000    0.0000         0    0.0000    0.0000
   -0.0000    0.0000   -0.0000    0.0000    0.3667   -0.0000         0    0.0000    0.0000
   -0.0000   -0.0000    0.0000    0.0000   -0.0000    2.4361         0    0.0000    0.0000
         0         0         0         0         0         0    2.6868         0         0
    0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000         0    2.9388   -0.0000
   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000         0   -0.0000    3.2931
```

b =

```
1.0e+03 *
```

0.0674	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	0	0.0000	-0.0000
0.0000	0.0694	-0.0000	-0.0000	0.0000	-0.0000	0	-0.0000	-0.0000
-0.0000	-0.0000	0.0709	0.0000	-0.0000	0.0000	0	-0.0000	0.0000
-0.0000	-0.0000	0.0000	0.0763	0.0000	0.0000	0	0.0000	0.0000
-0.0000	0.0000	-0.0000	0.0000	1.1334	-0.0000	0	0.0000	0.0000
-0.0000	-0.0000	0.0000	0.0000	-0.0000	7.3416	0	0.0000	0.0000
0	0	0	0	0	0	8.0937	0	0
0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	0	8.8497	-0.0000
-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000	0	-0.0000	9.9127

c =

1.0000	0.0000	0.0000	-0.0000	-0.0000	0.0000	0	-0.0000	-0.0000
0.0000	1.0000	0.0000	-0.0000	0.0000	-0.0000	0	-0.0000	-0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0	0.0000	0.0000
-0.0000	-0.0000	0.0000	1.0000	-0.0000	-0.0000	0	0.0000	0.0000
-0.0000	0.0000	0.0000	-0.0000	1.0000	-0.0000	0	0.0000	0.0000
0.0000	-0.0000	0.0000	-0.0000	-0.0000	1.0000	0	0.0000	0.0000
0	0	0	0	0	0	1.0000	0	0
-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000	0	1.0000	-0.0000
-0.0000	-0.0000	0.0000	0.0000	0.0000	0	0	-0.0000	1.0000

vi. Masukkan nilai K dan a₁ dengan menggunakan fungsi *listing*:

Kcap █ K+a §1.6

sehingga didapatkan

Kcap =

1.0e+06 *

0.0012	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	0	0.0000	-0.0000
0.0000	0.0020	-0.0000	0.0000	0.0000	-0.0000	0	-0.0000	-0.0000
-0.0000	-0.0000	0.0025	0.0000	-0.0000	0.0000	0	-0.0000	0.0000
-0.0000	0.0000	0.0000	0.0046	0.0000	0.0000	0	0.0000	0.0000
-0.0000	0.0000	-0.0000	0.0000	0.4126	-0.0000	0	0.0000	0.0000
-0.0000	-0.0000	0.0000	0.0000	-0.0000	2.8087	0	0.0000	0.0000
0	0	0	0	0	0	3.0990	0	0
0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	0	3.3907	-0.0000
-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000	0	-0.0000	3.8010

Langkah 8

Perhitungan untuk setiap langkah waktu i sampai batas sejumlah data yang telah ditentukan (sebanyak 2000 data dengan $\Delta t = 0.03$)

- g. Selesaikan persamaan untuk menentukan eksitasi pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
% Time step starts
for i = 1:1
    Pk = P(1:lambda,i+1) %2.1
    delP = Pk+a*depl(:,i)+b*vel(:,i)+c*accl(:,i) %2.1
```

Sehingga menghasilkan nilai eksitasi pada waktu $i=1$ adalah sebagai berikut:

delP =

1.0e+03 *

1.5517

0.0000

0.0116

-0.0032

0.0010

-0.0000

-0.0010

0.0002

-0.0002

- h. Selesaikan perpindahan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
depl(:,i+1) = Kcap\delP %2.2
```

Sehingga menghasilkan keluaran perpindahan modal pada waktu $i=1$ adalah sebagai berikut:

depl =

0.0040

0.0000

0.0003

-0.0000

0.0000

-0.0000

-0.0000

0.0000

-0.0000

- i. Selesaikan kecepatan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
|      vel(:,i+1) = a1*(depl(:,i+1)-depl(:,i))-a4*vel(:,i)-a6*accl(:,i)
```

Sehingga menghasilkan kecepatan modal pada waktu $i=1$ adalah sebagai berikut:

vel(:,2)

0.1326

0.0000

0.0111

-0.0014

0.0000

-0.0000

-0.0000

0.0000

-0.0000

- j. Selesaikan percepatan modal pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:

```
|      accl(:,i+1) = a2*(depl(:,i+1)-depl(:,i))-a3*vel(:,i)-a5*accl(:,i)
```

Sehingga menghasilkan percepatan modal pada waktu $i=1$ adalah sebagai berikut:

accl(:,2)

4.4201

0.0004

0.3716

-0.0475

0.0002

-0.0000

-0.0000

0.0000

-0.0000

- k. Selesaikan perpindahan total pada waktu $t=1$ dengan melakukan pemanggilan *listing* di bawah ini:


```

    U(:,i+1) = phi*depl(:,i+1) ; %2.5
end

```

Sehingga menghasilkan perpindahan total pada waktu $i=1$ adalah sebagai berikut:

U(:,2)

1.0e-05 *

-0.4440

-0.0632

-0.9047

-0.0377

-0.8715

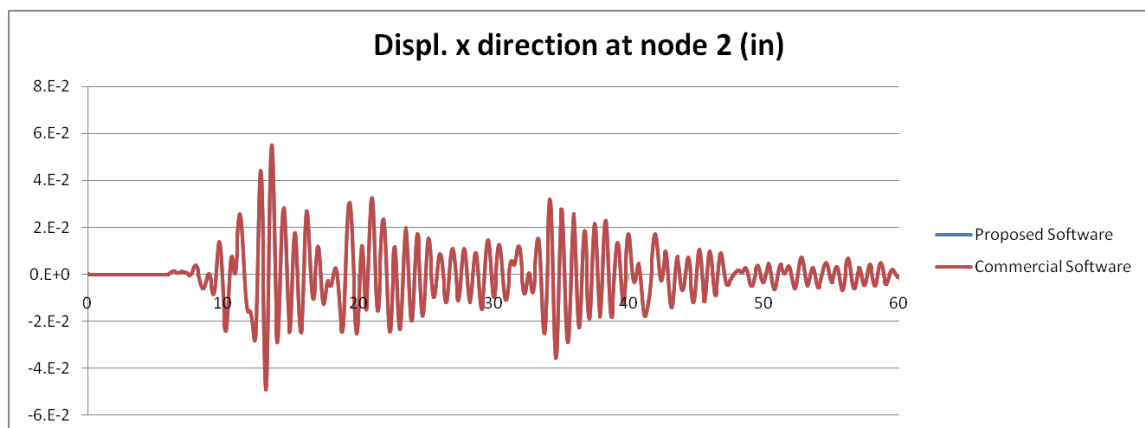
0.1136

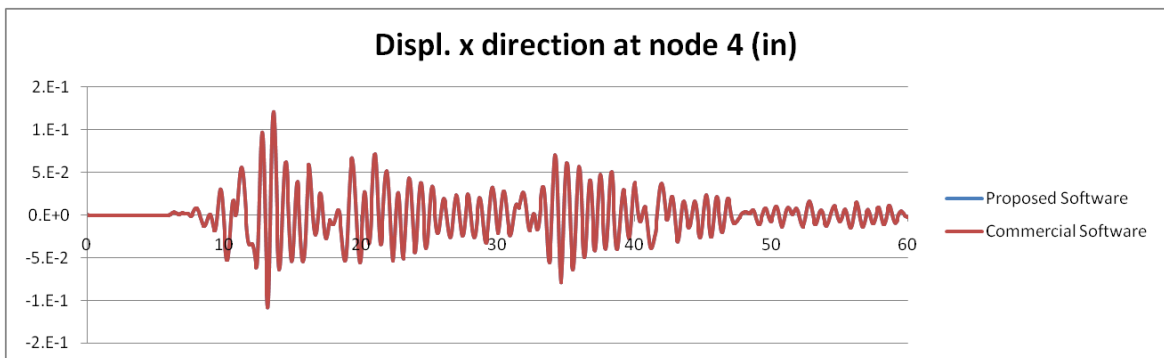
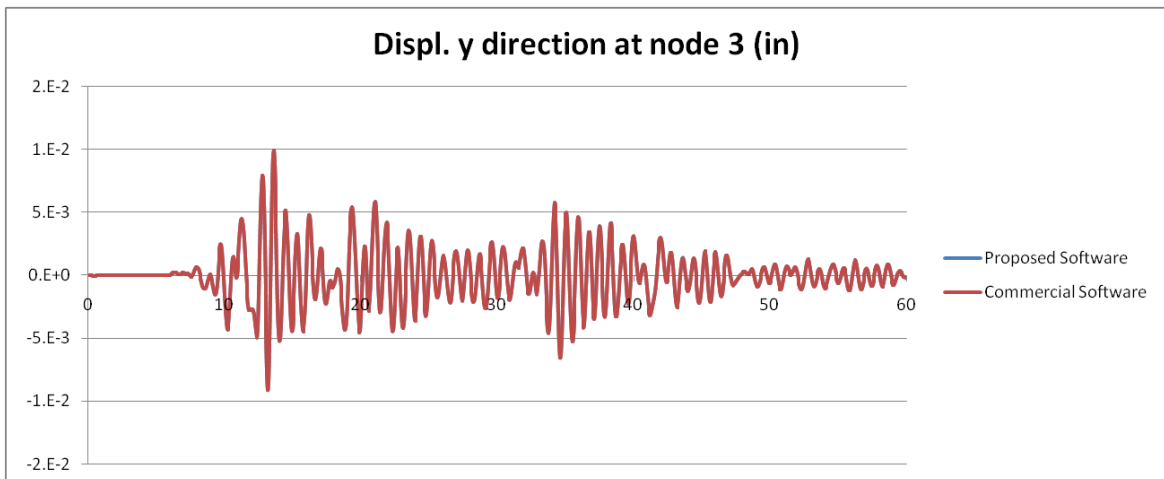
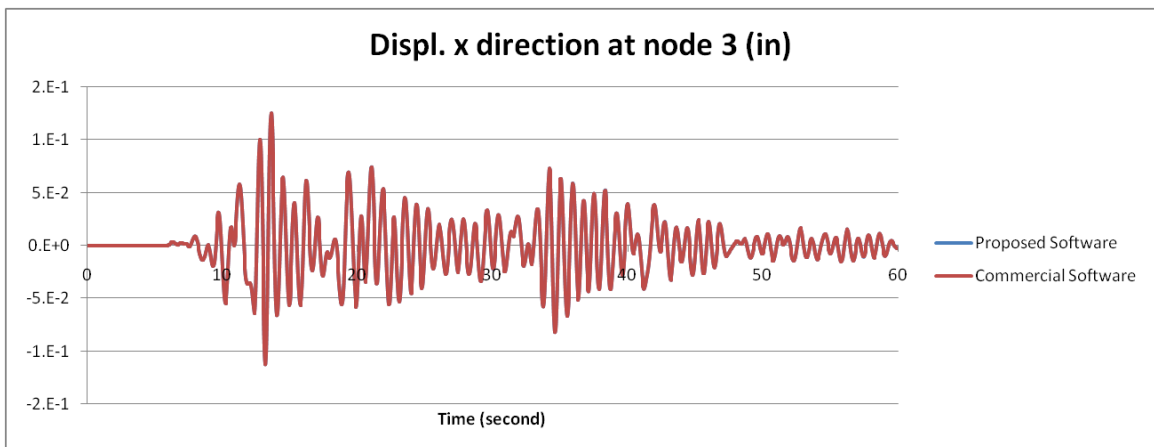
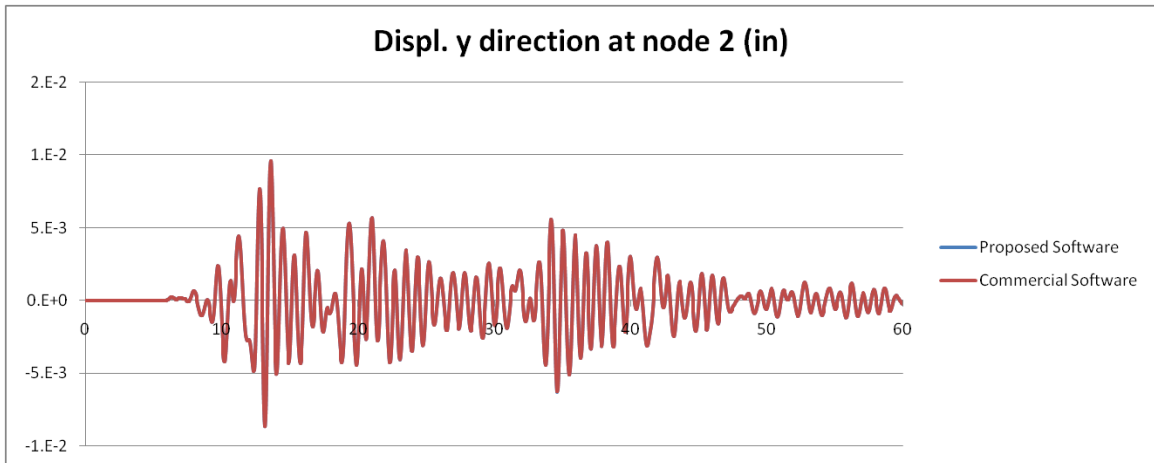
-0.3549

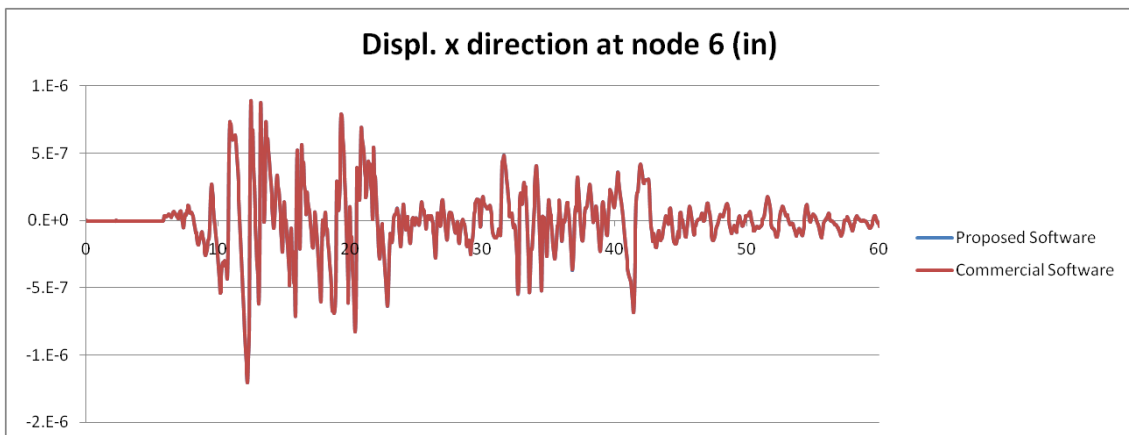
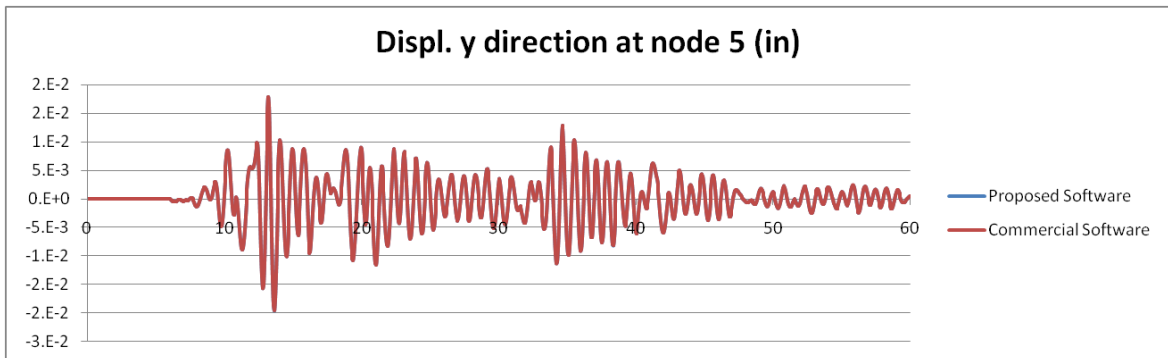
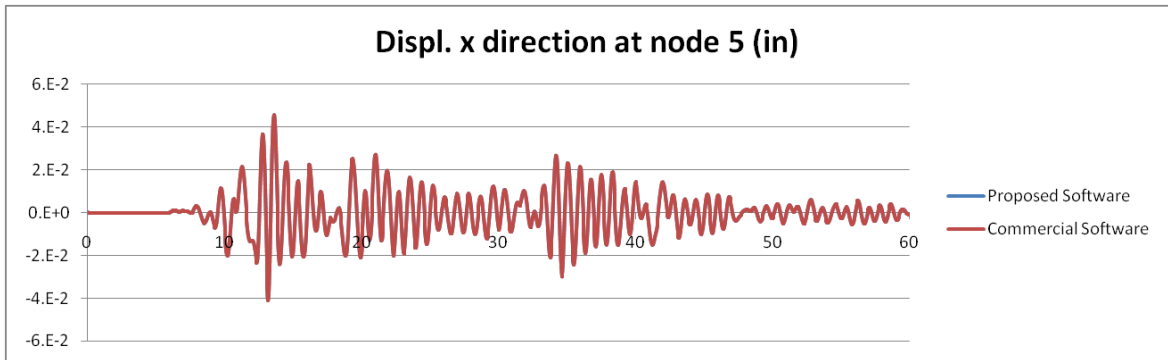
0.1015

-0.0003

1. Hasil dari perpindahan total dapat ditampilkan pada tabel dan di plotkan pada gambar berdasarkan fungsi waktu.
 - Hasil Grafik Displacement pada Matlab (*Proposed Software*) dan SAP 2000 (*Commercial Software*) sepanjang waktu respon percepatan gempa







(Halaman ini sengaja dikosongkan)

LAMPIRAN 2

INPUT DAN OUPUT PADA SOFTWARE

MCFEM MODEL DINAMIS

LAMPIRAN 2

INPUT DAN OUPUT PADA SOFTWARE MCFEM MODEL DINAMIS

a. Listing Fungsi Modul umum

i. Matriks Kekakuan

```
function y = PlaneTrussElementStiffness(E,A,L, theta)
%PlaneTrussElementStiffness This function returns the element
% stiffness matrix for a plane truss
% element with modulus of elasticity E,
% cross-sectional area A, length L, and
% angle theta (in degrees).
% The size of the element stiffness
% matrix is 4 x 4.
x = theta;
C = cosd(x);
S = sind(x);
y = E*A/L*[C*C C*S -C*C -C*S;
           C*S S*S -C*S -S*S;
           -C*C -C*S C*C C*S;
           -C*S -S*S C*S S*S] ;
```

ii. Matriks Massa

```
function y = PlaneTrussElementMass(rho,A,L, theta)
%PlaneTrussElementStiffness This function returns the mass
% matrix for a plane truss
% element with mass density rho,
% cross-sectional area A, length L, and
% angle theta (in degrees).
% The size of the element stiffness
% matrix is 4 x 4.
x = theta;
C = cosd(x);
S = sind(x);

%for lumped mass use the following
y=(rho*A*L/2)*[(C*C)+(S*S) (S*C)-(C*S) 0 0;
               (C*S)-(S*C) (C*C)+(S*S) 0 0;
               0 0 (C*C)+(S*S) (S*C)-(C*S);
               0 0 (C*S)-(S*C) (C*C)+(S*S)];
```

iii. Penggabungan Matriks

```
function y = PlaneTrussAssemble(K,k,i,j)
%PlaneTrussAssemble This function assembles the element stiffness
% matrix k of the plane truss element with nodes
% i and j into the global stiffness matrix K.
% This function returns the global stiffness
% matrix K after the element stiffness matrix
% k is assembled.
lm(1)=2*i-1;
lm(2)=2*i;
lm(3)=2*j-1;
lm(4)=2*j;
for m=1:4
    ii=lm(m);
```

```

    for n=1:4
        jj=lm(n);
        K(ii,jj)=K(ii,jj)+k(m,n);
    end
end
y = K;

```

b. *Plane Truss* Model Segitiga

Listing fungsi:

```

clc;clear;

load s1000

xc = 0 ;
x = 1 ;
hitung = 0 ;
simulasigagal = 0 ;
d02x = 0 ;
d02z = 0 ;

for i=1:x

    % Properti Finite Element
    K = zeros(6,6) ;
    M = zeros(6,6) ;

    % Baja A36
    e = abs(normrnd(29*10^6,1740000)) ;
    rho = 7.345E-04 ;
    d = 13.6 ;
    r = d/2 ;
    csa = r^2*pi() ;
    l1 = 118.11 ;
    l2 = 167.0328 ;

    % Global matrix calculation of element
    k1 = PlaneTrussElementStiffness(e,csa,l1,90) ;
    k2 = PlaneTrussElementStiffness(e,csa,l2,315) ;
    m1 = PlaneTrussElementMass(rho,csa,l1,90) ;
    m2 = PlaneTrussElementMass(rho,csa,l2,315) ;

    % Assemble matrix element
    K = PlaneTrussAssemble(K,k1,1,2) ;
    K = PlaneTrussAssemble(K,k2,2,3) ;

    M = PlaneTrussAssemble(M,m1,1,2) ;
    M = PlaneTrussAssemble(M,m2,2,3) ;

    format long ;

    Madd = abs(normrnd(100000,10000)) ;

    % Apply boundary conditions for stiffness matrix and mass
    K = K(3:4,3:4) ;
    M = M(3:4,3:4)+diag([Madd;Madd]) ;
    % sdof = length(K);

```

```

-----%
%-----%

% Transforming the System Matrices to modal coordinates
lambda = 2 ; % Number of modes to be considered

% Modal Analysis
[V,D]=eig(K,M) ;

Factor=diag(V'*M*V) ;

Phi=V*inv(sqrt(diag(Factor))) ; %normalisasi mode shape
Omega=diag(sqrt(Phi'*K*Phi)) ; %cek orthogonal

% Selecting only first two Natural Frequencies and Mode shapes
w = diag(Omega(1:lambda)) ; %periode natural sebenarnya
phi = Phi(:,1:lambda) ; %moda getar sebenarnya

% Matriks K dan M bentuk Tergeneralisasi
MM = phi'*M*phi ;
K = phi'*K*phi ;

% Damping
aa = [1/Omega(1) Omega(1);1/Omega(2) Omega(2)] ;
bb = inv(aa)*2*[0.05;0.05] ;
z = bb(1)*M+bb(2)*K ;
C = phi'*z*phi ;

-----%
%-----%

% Pemilihan Jenis Percepatan
acceleration = 'Average' ;

switch acceleration
    case 'Average'
        gaama = 1/2 ;beta = 1/4 ;
    case 'Linear'
        gaama = 1/2 ;beta = 1/6 ;
end

% Time step (dari sekian banyak data)
ti = 0 ;
tf = 60 ;
dt = 0.06 ;
t = ti:dt:tf ;
nt = fix((tf-ti)/dt) ;
n = length(M) ;

% Excitation
H = [1;0] ;
p = -M*H*g8' ;
P = phi'*p ;
s = [g8'] ;

depl = zeros(n,nt) ;

```



```

vel = zeros(n,nt) ;
accl = zeros(n,nt) ;
U = zeros(n,nt) ;

% Initial Conditions
depl(:,1) = zeros ;
vel(:,1) = zeros ;
U(:,1) = phi*depl(:,1) ;
P0 = zeros(n,1) ;

accl(:,1) = MM\ (P0-C*vel(:,1)-K*depl(:,1)) ; %1.3

% Constants used in Newmark's integration
a1 = gaama/(beta*dt) ; a2 = 1/(beta*dt^2) ; %a1
a3 = 1/(beta*dt) ; a4 = (gaama/beta)-1 ; %a2
a5 = (1/(2*beta))-1 ; a6 = dt*((gaama/(2*beta))-1) ; %a3

a = a1*C+a2*MM ; %a1
b = a3*MM+a4*C ; %a2
c = a5*MM+a6*C ; %a3
Kcap = K+a ; %1.6

% Keandalan
Ucontrol = 0.161281241*ones(1,2000) ;

% Time step starts
for i = 1:nt
    Pk = P(1:lambda,i+1) ; %2.1
    delP = Pk+a*depl(:,i)+b*vel(:,i)+c*accl(:,i) ; %2.1
    depl(:,i+1) = Kcap\delP ; %2.2
    vel(:,i+1) = a1*(depl(:,i+1)-depl(:,i))-a4*vel(:,i)-
a6*accl(:,i) ; %2.3
    accl(:,i+1) = a2*(depl(:,i+1)-depl(:,i))-a3*vel(:,i)-
a5*accl(:,i) ; %2.4
    U(:,i+1) = phi*depl(:,i+1) ; %2.5
    u1 = abs(U(1,i+1)) ;
    u2 = abs(U(2,i+1)) ;
    if u1 >= Ucontrol
        hitung=hitung+1 ;
    end
    d02x = d02x+u1 ;
    d02z = d02z++u2 ;
    xc=xc+1 ;
end

%-----
-%
struktur_gagal = hitung/nt ;

if struktur_gagal > 0
    simulasi_gagal = simulasi_gagal+1 ;
end
end

rata_rata_d2x = d02x/(x*nt)
rata_rata_d2z = d02z/(x*nt)
Pf = (simulasi_gagal/x)

```

```
zz=max (U')
```

Output dengan 150.000 kali simulasi:

rata_rata_d2x =

0.015458944448145

rata_rata_d2z =

0.004843247833421

Pf =

0

c. *Plane Truss* Struktur 2 Tingkat

Listing fungsi:

```
clc;clear;

load s1000

xc = 0 ;
x = 1 ;
hitung = 0 ;
simulasigagal = 0 ;
d03x = 0 ;
d03z = 0 ;

for i=1:x
    % Properti Finite Element
    K = zeros(12,12) ;
    M = zeros(12,12) ;

    % Baja A36
    e = abs(normrnd(29*10^6,1740000)) ;
    rho = 7.345E-04 ;
    d = 30 ;
    r = d/2 ;
    csa = r^2*pi() ;
    l1 = 118.11 ;
    l2 = 167.0328 ;

    % Global matrix calculation of element
    k1 = PlaneTrussElementStiffness(e,csa,l1,0) ; % k1,1,6
    m1 = PlaneTrussElementMass(rho,csa,l1,0) ;
    k2 = PlaneTrussElementStiffness(e,csa,l1,90) ; % k2,1,2
    m2 = PlaneTrussElementMass(rho,csa,l1,90) ;
    k3 = PlaneTrussElementStiffness(e,csa,l2,45) ; % k3,1,5
    m3 = PlaneTrussElementMass(rho,csa,l2,45) ;
    k4 = PlaneTrussElementStiffness(e,csa,l1,270) ; % k4,5,6
    m4 = PlaneTrussElementMass(rho,csa,l1,270) ;
```

```

k5 = PlaneTrussElementStiffness(e,csa,11,0) ; % k5,2,5
m5 = PlaneTrussElementMass(rho,csa,11,0) ;
k6 = PlaneTrussElementStiffness(e,csa,11,90) ; % k6,2,3
m6 = PlaneTrussElementMass(rho,csa,11,90) ;
k7 = PlaneTrussElementStiffness(e,csa,12,45) ; % k7,2,4
m7 = PlaneTrussElementMass(rho,csa,12,45) ;
k8 = PlaneTrussElementStiffness(e,csa,11,270) ; % k8,4,5
m8 = PlaneTrussElementMass(rho,csa,11,270) ;
k9 = PlaneTrussElementStiffness(e,csa,11,0) ; % k9,3,4
m9 = PlaneTrussElementMass(rho,csa,11,0) ;

% Assemble matrix element
K = PlaneTrussAssemble(K,k1,1,6) ;
K = PlaneTrussAssemble(K,k2,1,2) ;
K = PlaneTrussAssemble(K,k3,1,5) ;
K = PlaneTrussAssemble(K,k4,5,6) ;
K = PlaneTrussAssemble(K,k5,2,5) ;
K = PlaneTrussAssemble(K,k6,2,3) ;
K = PlaneTrussAssemble(K,k7,2,4) ;
K = PlaneTrussAssemble(K,k8,4,5) ;
K = PlaneTrussAssemble(K,k9,3,4) ;

M = PlaneTrussAssemble(M,m1,1,6) ;
M = PlaneTrussAssemble(M,m2,1,2) ;
M = PlaneTrussAssemble(M,m3,1,5) ;
M = PlaneTrussAssemble(M,m4,5,6) ;
M = PlaneTrussAssemble(M,m5,2,5) ;
M = PlaneTrussAssemble(M,m2,2,3) ;
M = PlaneTrussAssemble(M,m3,2,4) ;
M = PlaneTrussAssemble(M,m4,4,5) ;
M = PlaneTrussAssemble(M,m5,3,4) ;

format long ;

Madd = abs(normrnd(100000,10000)) ;

% Apply boundary conditions for stiffness matrix and mass
K = K(3:11,3:11) ;
M = M(3:11,3:11) + diag([0;0;Madd;Madd;Madd;Madd;0;0;0]) ;
% sdof = length(K);

%-----
-%

% Transforming the System Matrices to modal coordinates
lambda = 9 ; % Number of modes to be considered

% Modal Analysis
[V,D]=eig(K,M) ;

Factor=diag(V'*M*V) ;

Phi=V*inv(sqrt(diag(Factor))) ; %normalisasi mode shape
Omega=diag(sqrt(Phi'*K*Phi)) ; %cek orthogonal

% Selecting only first two Natural Frequencies and Mode shapes
w = diag(Omega(1:lambda)) ; %frekuensi natural sebenarnya

```

```

phi = Phi(:,1:lambda) ; %moda getar sebenarnya

% Matriks K dan M bentuk Tergeneralisasi
MM = phi'*M*phi ;
KK = phi'*K*phi ;

% Damping 5%
aa = [1/Omega(1) Omega(1);1/Omega(2) Omega(2)] ;
bb = inv(aa)*2*[0.05;0.05] ;
z = bb(1)*M+bb(2)*K ;
C = phi'*z*phi ;

%-----
-%

% Pemilihan Jenis Percepatan
acceleration = 'Average' ;

switch acceleration
case 'Average'
    gaama = 1/2 ;beta = 1/4 ;
case 'Linear'
    gaama = 1/2 ;beta = 1/6 ;
end

data) % Time step (tergantung kita mendefinisikan dari sekian banyak
ti = 0 ;
tf = 60 ;
dt = 0.06 ;
t = ti:dt:tf ;
nt = fix((tf-ti)/dt) ;
n = length(MM) ;

% Excitation
H = [1;0;1;0;1;0;1;0;1];
p = -M*H*g8' ;
P = phi'*p ;
s = [g8'] ;

depl = zeros(n,nt) ;
vel = zeros(n,nt) ;
accl = zeros(n,nt) ;
U = zeros(n,nt) ;

% Initial Conditions
depl(:,1) = zeros ;
vel(:,1) = zeros ;
U(:,1) = phi*depl(:,1) ;
P0 = zeros(n,1) ;

accl(:,1) = MM\ (P0-C*vel(:,1)-KK*depl(:,1)) ; %1.3

% Constants used in Newmark's integration
a1 = gaama/(beta*dt) ; a2 = 1/(beta*dt^2) ; %a1
a3 = 1/(beta*dt) ; a4 = (gaama/beta)-1 ; %a2
a5 = (1/(2*beta))-1 ; a6 = dt*((gaama/(2*beta))-1) ; %a3

```

```

a = a1*C+a2*MM ; %a1
b = a3*MM+a4*C ; %a2
c = a5*MM+a6*C ; %a3
Kcap = KK+a ; %1.6

Ucontrol = 0.161281241*ones(1,2000) ;

% Time step starts
for i = 1:nt
    Pk = P(1:lambda,i+1) ; %2.1
    delP = Pk+a*depl(:,i)+b*vel(:,i)+c*accl(:,i) ; %2.1
    depl(:,i+1) = Kcap\delP ; %2.2
    vel(:,i+1) = a1*(depl(:,i+1)-depl(:,i))-a4*vel(:,i)-
a6*accl(:,i) ; %2.3
    accl(:,i+1) = a2*(depl(:,i+1)-depl(:,i))-a3*vel(:,i)-
a5*accl(:,i) ; %2.4
    U(:,i+1) = phi*depl(:,i+1) ; %2.5
    u3x = abs(U(3,i+1)) ;
    u3z = abs(U(4,i+1)) ;
    if u3x >= Ucontrol
        hitung=hitung+1 ;
    end
    d03x = d03x+u3x ;
    d03z = d03z+u3z ;
    xc=xc+1 ;
end

%-----
-%

struktur_gagal = hitung/nt ;

if struktur_gagal > 0
    simulasi_gagal = simulasi_gagal+1 ;
end

end

rata_rata_d3x = d03x/(x*nt)
rata_rata_d3z = d03z/(x*nt)
Pf = (simulasi_gagal/x)

zz = max([U(3,:) ' U(4,:) '])

```

Output dengan 150.000 kali simulasi:

rata_rata_d3x =

0.017829864467202

rata_rata_d3z =

0.001411170619626

Pf =

0

BIODATA PENULIS

BIODATA PENULIS



Muhamamd Ichsan lahir di Sulawesi Selatan 25 September 1995, anak ke dua dari pasangan Nasrullah Hafid dan Dian Ekawati. Mulai menempuh Pendidikan pada SDN 006 Kampung Baru kemudian pindah ke SDN 051 Inpres Lampa, MTs DHI Guppi Mapilli yang berada di Sulawesi Barat, kemudian melanjutkan SMAN 2 Tinggimoncong di Sulawesi Selatan. Melalui jalur SNMPTN Undangan, penulis dapat melanjutkan pendidikan tinggi di Teknik Kelautan ITS Surabaya. Selama kuliah, penulis aktif di organisasi Lembaga

Dakwah Kampus (LDK) JMMI, Lembaga Dakwah Fakultas (LDF) Al Bahri dan Lembaga Dakwah Jurusan (LDJ) Bahrul ‘Ilmi Teknik Kelautan. Selama aktif di LDK, penulis pernah menjabat sebagai staf magang di departemen Kaderisasi Jamaah Masjid Manarul Ilmi (JMMI) ITS pada tahun pertama sekaligus staf di LDJ pada departemen Kaderisasi, pada tahun ke dua penulis menjadi kepala Departemen Kaderisasi LDJ kepengurusan 2013/2014 sekaligus staf pada Departemen Syi’ar di LDF, dan pada tahun ke tiga menjadi Kepala Departemen Kaderisasi LDJ untuk kepengurusan 2014/2015.

Selain memiliki pengalaman organisasi, penulis juga meluangkan waktu untuk meningkatkan kemampuan akademik dan dunia kerja, selain melalui kerja praktek pada Uji Resisten Kapal di BPT Hidrodinamika Indonesia dan Departemen *Pipeline* PT. ZEE Indonesia penulis juga meluangkan waktu untuk magang di Divisi *Aerostructure* PT. Dirgantara Indonesia.

Penulis juga menjadi *grader* mata kuliah Pemrograman Komputer. Pada bulan November hingga Desember tahun 2017, penulis menyelesaikan Tugas Akhir sebagai syarat kelulusan Strata-1, dengan topik “Pengembangan Software Analisa Keandalan Struktur dengan Model Dinamis Berbasis Monte Carlo Finite Element Method pada Struktur Sederhana 2 Dimensi”.

Email:

Muhammadichsan1995@gmail.com